CANDIDATE SAMPLING SCHEMES AND SOME IMPORTANT APPLICATIONS

By

BRIAN S. CAFFO

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2001

I dedicate this work to Kerri.

ACKNOWLEDGMENTS

Foremost I would like to thank my advisor and friend, Jim Booth, for all his help both in my dissertation and throughout graduate school. I would also like to thank Kerri, my mother, my father, Steve and Theresa. Their unconditional love and support made this dissertation possible. I owe a great deal of thanks to Alan Agresti for all of his advice, mentoring and funding for the past three years. I also appreciate the help of my three remaining committee members, Jim Hobert, Scott McCullough and Andy Rosalsky. Selecting my committee was easy. They were the five best teachers I had in graduate school. Though we have never met in person, I would like to thank Anthony Davison for working with me via emails and faxes. I would also like to thank the faculty of the Department of Statistics, especially Malay Ghosh, Ron Randles, Jim Kepner, Brett Presnell and Dennis Wackerly. I should also recognize my grandparents, Betty, Louis, Emily and Bud, for always being a source of inspiration for me.

The people who helped me the most through this process were my classmates and friends. A short and incomplete list of people I would like to give special thanks to is the following: Galin, Wolfgang, Ziyad, Allen, Chad, Patches, Jeffy, Jamie, Matt, Susan, Thomas and Heather for all of their encouragement and help. Finally I would like to thank our zoo, Mugsy, Kid Whiskers, Milo, Bishop and Gill, for reminding me that a sunny spot to nap in is way, way more important than writing computer code.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

CANDIDATE SAMPLING SCHEMES AND SOME IMPORTANT APPLICATIONS

By

Brian S. Caffo

August 2001

Chair: James G. Booth
Major Department: Statistics

Monte Carlo and Markov chain Monte Carlo approximate intractable expectations with respect to a distribution F by applying the strong law of large numbers to averages of either independent, identically distributed F variates or Markovian variates possessing F as their stationary distribution. Under regularity conditions, central limit theorems can be used to quantify the error between these averages and the expectations being estimated. A minimal requirement to perform Monte Carlo or Markov chain Monte Carlo is an algorithm to either generate F variates or a Markov chain with F as its stationary distribution. Accept/reject sampling and the Metropolis Hastings algorithm are two such methods that are useful when F is difficult to simulate from but a dominating distribution or transition kernel, G, is easy to simulate from. The application and study of these two methods are the primary emphases for this dissertation.

In this dissertation we propose an application of the Metropolis Hastings algorithm to *Monte Carlo conditional inference*. For contingency tables, Monte Carlo conditional inference reduces to generating vectors of non-negative integers satisfying linear constraints according to a generalized hypergeometric distribution. Our algorithm uses a rounded normal approximation to the generalized hypergeometric distribution as the dominating measure for the Metropolis Hastings algorithm. The algorithm creates a Markov chain that updates a few cells of the current state of the chain at each iteration.

We propose a second algorithm that modifies accept/reject sampling. This modification circumvents calculating $\sup dF/dG$ by replacing it with the largest observed value of dF/dG. We verify the theoretical validity of this modification and test the algorithm on several examples. We particularly focus on its application to the Monte Carlo EM (MCEM) algorithm. MCEM replaces intractable expectations in the "E" step of the EM algorithm with a Monte Carlo approximation. In this dissertation we illustrate how this can be done using our modification of accept/reject sampling. Further, as the "E" step of the MCEM algorithm often factors into the sum of expectations of independent random variables, we suggest an improvement of the MCEM algorithm that optimally allocates the Monte Carlo resources among the expectations.

CHAPTER 1
INTRODUCTION

1.1 Introduction

Consider the problem of evaluating the integral

$$\int_{\mathcal{X}_F} h(x)dF(x) \equiv \mu_h, \tag{1.1}$$

where F is a known distribution with support $\mathcal{X}_F$ and h is a real valued function. In many practical situations calculating this integral exactly is impossible due to the complexity of F (or perhaps the size of $\mathcal{X}_F$ in the discrete case). The *Monte Carlo* solution to this problem mirrors what a statistician would do if F was unknown; that is, collect data from F and then use it to estimate μ_h. Monte Carlo methods have enjoyed an incredible increase in popularity over the past two decades (see Figure 1.1). The exponential growth in popularity can be attributed to increases in computing power allowing for more and more situations where Monte Carlo methods can be successfully implemented when other approximations fail. Included in these situations are cases where possibly accurate numerical approximations are considered suspect because no estimate of the error is available. In contrast, the error associated with nearly all methods of Monte Carlo approximation can be assessed using probabilistic methods.

A minimal requirement for Monte Carlo approximation is the ability to simulate from F. However, direct simulation from F by applying the probability integral transform to simulated random uniforms (Robert and Casella; 1999, page 36) is often not possible for the same reason Equation (1.1) is intractable. It is also often the case that there is no transformation of easy to simulate random variables

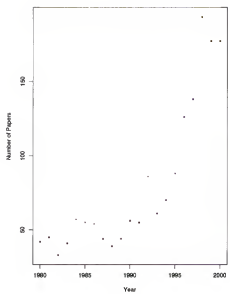

Figure 1.1: Number of papers in statistics journals with "Monte Carlo" in their key words or title by year.

that produces F variates. Candidate sampling schemes, such as accept/reject sampling (Devroye; 1986) and the Metropolis Hastings algorithm (Hastings; 1970) avoid this problem by simulating from a dominating distribution, say G, and accepting those simulated variates most consistent with F. For both algorithms, the measure of this consistency with F depends on the ratio of the densities or mass functions. The application and theory of these candidate sampling schemes are the primary emphases of this dissertation.

As a point of departure, we assume the random numbers the computer generates are actually distributed as uniform(0,1). Of course, in reality, computer generated random numbers may take only finitely many values and are not random at all. However, good random number generators will produce output that passes various classes of statistical tests of randomness and uniformity for extremely large sample sizes. By this criterion, the three methods of random uniform generation used for this dissertation, Park and Miller (1988), L'Ecuyer

(1997) and Marsaglia and Zaman (1994), are all very good. To present evidence to this effect we calculated the Kolmogorov-Smirnov test for the default random uniforms (Park and Miller; 1988) generated by the programming language Ox (http://www.nuff.ox.ac.uk/Users/Doornik/index.html), for as large a sample size as possible. The asymptotic P-value (derivations given in Feller; 1948) for a sample size of ten million was .974. Thus, on the basis of this test, there is no evidence to suggest Ox's default generated random numbers are anything other than uniform.

For the remainder of this introductory chapter, we outline the main topics in the dissertation. Although the topics are somewhat disjointed, they share the common thread of either implementing or studying a candidate sampling scheme. The most general application studied is a modification of accept/reject simulation, which we now outline in the next section.

1.2 Accept/Reject Sampling

As mentioned in the introduction, the first step in Monte Carlo estimation of Equation (1.1) is the ability to simulate from the *target* distribution F. Accept/reject sampling is a way to generate an independent identically distributed (i.i.d.) sequence from F by thinning out an i.i.d. sequence from a dominating distribution that is easy to simulate from. We illustrate rejection sampling with a simple example. Consider simulating from the semicircular density, $f(x) = \sqrt{(a/2)^2 - (x - a/2)^2}$ for $0 \leq x \leq a$, where $a = 2\sqrt{2/\pi}$. One accept/reject sampler for this density consist of picking a point at random from the square with corners $(0, 0)$ and (a, b) for some $b \geq a/2$, and accepting those that lie under $f(x)$. The sequence of accepted X-coordinates is then an i.i.d. sample from F (where, recall, F is the distribution function associated with f). To illustrate, consider the two random points in Figure 1.2. Accept/reject sampling would accept the point $(X2, Y2)$ and reject the point $(X1, Y1)$. The probability a point is accepted is the ratio of the area of the semi-circle to the area of the square, or $1/ab$. Further, the

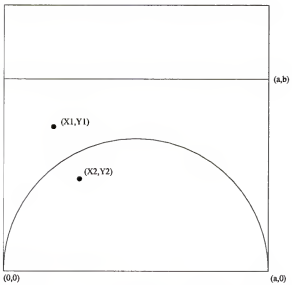

Figure 1.2: A simple accept/reject sampler.

probability that a point (X, Y) is accepted, *and* $X \leq x$ (for $0 < x < a$) is the ratio of the area of the shaded region in Figure 1.3 and ab, or

$$\int_0^x f(t)dt/ab = F(x)/ab.$$

It follows that the probability an accepted X is less than x is $F(x)$ as required. Notice that the acceptance rate, $1/ab$, is largest when b is as small as possible, i.e. when $b = a/2 = \sqrt{2/\pi}$ giving an acceptance rate of $\pi/4$ (roughly 79%).

More generally, either the support or the range of the target density will be unbounded in which case candidate variates cannot be chosen uniformly. General accept/reject sampling requires the existence and discovery of the finite supremum $C \equiv \sup_x f(x)/g(x)$ over the support of the target where g is the candidate density. It is often the case that calculating this upper bound exactly is difficult or impossible for otherwise desirable candidate distributions. In Chapter 5 we develop a method for circumventing the computation of C by estimating it with a sequence of lower bounds obtained as the maximum observed value of f/g from the simulated candidates. This maximum, or empirical supremum (ESUP), is

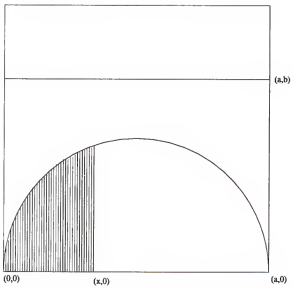

Figure 1.3: Accept/reject diagram, the shaded area is $F(x)$.

a superefficient estimator of C. We exploit its fast rate of convergence to show that rejection sampling using the empirical supremum accepts essentially the same candidates as rejection sampling with a known supremum (KSUP rejection sampling).

1.3 MCEM

The EM algorithm (Dempster et al.; 1977; Robert and Casella; 1999) is an algorithm for maximizing the marginal log-likelihood in missing data problems. The EM algorithm progresses by maximizing the expected complete data log-likelihood[1] with respect to the distribution of the missing data given both the observed data and the previous parameter estimates. A Monte Carlo version of EM (Wei and Tanner; 1990), replaces the expectation, or "E" step, with a Monte Carlo estimate and is referred to as the MCEM algorithm.

[1] The complete data log-likelihood is the log of the joint density of the observed and missing data.

Table 1.1: Cross-classified tumor ratings of two pathologists.

	Pathologist B				
Pathologist A	1	2	3	4	5
1	22	2	2	0	0
2	5	7	14	0	0
3	0	2	36	0	0
4	0	1	14	7	0
5	0	0	3	0	3

Source: Landis and Koch (1977)

It is often the case that the E step is the sum of integrals arising from independent clusters. MCEM practitioners typically estimate each of these integrals with the same Monte Carlo sample size. However, we argue in Chapter 6 that such allocation is suboptimal, as some clusters can be more influential on parameter estimates than others. To exploit this, we present an optimal allocation rule that comes at no added computational cost. Two examples are given in which optimal allocation reduces the Monte Carlo variance of the parameter estimates by half or more.

1.4 Exact Conditional Tests

Chapters 3 and 4 deal with *exact conditional tests*. We introduce exact conditional tests for log-linear models here, with further specifics in Chapter 3. Table 1.1 shows the cross-classified ratings of tumors by two pathologists (Landis and Koch; 1977). Consider the following seemingly simple task: given the 5×5 table of non-negative integers, list all tables with the same margins.

It is relatively straightforward to write down an algorithm that cycles through all possibilities exactly once. Surprisingly, there are over 12 billion tables that satisfy these margins, making enumeration unreasonably time consuming by current computing standards. The difficulty lies in the fact that the "margin problem" is np-complete. That is, the number of operations required grows faster than any polynomial in the size of the table. No increases in processor speed in

the foreseeable future will make this problem tractable for even moderate table dimensions. Now consider a slight modification to the problem: list all tables with the same margins and the same value of the sum $\sum_{i,j} y_{ij} ij$, where y_{ij} is the $(ij)th$ table entry. It turns out there are fewer than 35,000 such tables. However, there is no obvious algorithm to efficiently cycle through tables satisfying the extra constraint. Clearly an *enumerate and reject* strategy of enumerating the 12 billion tables to count the 35,000 would be very inefficient.

These two problems can be recast in the following more general framework. Find the set $\Gamma = \{\mathbf{y} \in \mathbb{Z}_+^{25} | \mathbf{X}^t\mathbf{y} = \mathbf{s}\}$ where $\mathbf{y}$ is a vector representing the 25 table entries, $\mathbf{X}^t$ is a $p \times 25$ matrix with values in $\mathbb{Z}_+$, and $\mathbf{s}$ is the value of $\mathbf{X}^t\mathbf{y}$ for the initial table. Here $\mathbb{Z}_+$ represents the set of non-negative integers. We suppose that the vector $\mathbf{y}$ consists of the y_{ij} values listed lexicographically according to their subscripts, i.e. $\mathbf{y} = (y_{11}, y_{12}, \ldots, y_{21}, y_{22}, \ldots, y_{55})^t$. Thus for the margin problem $\mathbf{X}^t$ is given by

$$\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}. \tag{1.2}$$

The second problem requires the inclusion of the additional row,

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 2 & 4 & 6 & \ldots & 25 \end{pmatrix}.$$

The mathematical problem of enumerating Γ is equivalent to the computational issues involved in the conditional analysis of log-linear models for categorical data. To be specific, $\mathbf{Y}$ represents the data consisting of independent Poisson random variables with mean vector $\boldsymbol{\mu}$. If $\boldsymbol{\mu}$ satisfies $\log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta}$ (where log acts component-wise on vectors) then $\mathbf{X}^t\mathbf{Y} = \mathbf{S}$ is complete sufficient for $\boldsymbol{\beta}$. Recall, the distribution obtained by conditioning on the full sufficient statistic does not depend on any parameters. For this reason, conditional inference eliminates $\boldsymbol{\beta}$ by considering the distribution of $\mathbf{Y}$ given $\mathbf{S}$ (denoted $\mathbf{Y}|\mathbf{S} = \mathbf{s}$). This conditional distribution has the reference set, Γ, as its support. As an example, the $\mathbf{X}^t$ matrix given by Equation (1.2) represents a model that assumes independence between rows and columns. In this case, $\mathbf{Y}|\mathbf{S} = \mathbf{s}$ with $\mathbf{s}$ equal to the observed margins is the multivariate non-central hypergeometric distribution. Adding the extra row to $\mathbf{X}$ corresponds to a *uniform association* model.

In the previous example, the independence model holds for any value of $\boldsymbol{\beta}$. Therefore, $\boldsymbol{\beta}$ is not of interest when testing the independence assumption. Such parameters that are not of primary interest are called nuisance parameters. Conditioning on the appropriate sufficient statistics allows inferences to be made that are free of nuisance parameters. Although conditioning in this manner can be applied more generally, we restrict attention to log-linear models for Poisson and multinomial data in this dissertation. Surveys of conditional tests for categorical data are given in Agresti (1992) and Agresti (2001) while conditional methodology in general is reviewed by Reid (1995).

Exact probability calculations from the distribution of $\mathbf{Y}|\mathbf{S}$ for the purpose of inference require knowledge of the reference set. As discussed earlier, the size or complexity (or both) of the reference set Γ often makes enumeration impossible. Monte Carlo methods offer a viable alternative provided one can simulate from the distribution of $\mathbf{Y}|\mathbf{S}$. In Chapter 3 we review enumeration and simulation

techniques for exact conditional testing, pointing out the limitations of current methodology. In Chapter 4 we present a new method for generating a dependent (Markovian) sample from the conditional distribution. We apply this procedure to examples where no other current methods apply.

1.5 Outline

The remainder of the dissertation is organized as follows. Chapter 2 briefly reviews common Monte Carlo and Markov chain Monte Carlo algorithms. The primary emphasis is covering algorithms that are needed later. Chapter 3 reviews exact conditional analysis, emphasizing log-linear models. Though none of this material is new research, the treatment is novel and associations are made between areas that were previously unconnected to the best of our knowledge. Chapter 4 presents an MCMC algorithm for performing exact conditional tests for log-linear models. The material from this chapter is accepted for publication in the *Journal of Computational and Graphical Statistics*. In Chapter 5 we present our modification of accept/reject simulation. In this chapter we formally prove that the accepted variables may be used as if they were independent and identically distributed, even though they are neither. We illustrate this with some examples, including a Bayesian model and an exact conditional test. This chapter is taken from the joint work of Caffo, Booth and Davison (2001). In Chapter 6 we discuss the MCEM algorithm. We first present a rule for allocating Monte Carlo resources in the MCEM algorithm. We illustrate this rule with a beta-binomial model. The content of the optimal allocation sections are to appear in *Computational Statistics and Data Analysis*. We then analyze a logit/normal GLMM using the MCEM algorithm and ESUP rejection sampling. We further show that the numerical maximization used in the Laplace approximation for the MCEM algorithm can be circumvented. It is important to note that all of the MCEM computing tricks in

Chapter 6 come at essentially zero extra calls to the cpu. Finally the dissertation concludes with a discussion of future research in Chapter 7.

Although some notational changes are necessary in different chapters, the following conventions are always used. The target density and distribution are always labeled some variation of f and F respectively while the candidate (transition) density and distribution are always labeled g and G. When necessary, expectations and probabilities with respect to F and G will be denoted E_F and E_G respectively. Random variables are always in capitals (such as Y) and observed values, arguments of a density or distribution function, etc. are the corresponding lowercase letter (for example, y). Variables that are vectors or matrices are written in bold (such as $\mathbf{y}$) while variables that are scalars or are left undefined (such as when discussing Monte Carlo in generality) are not. A superscript t is used to denote transpose (for example, $\mathbf{X}^t$ is the transpose of $\mathbf{X}$) while a $'$ is used to denote differentiation. The letter (uppercase italic) C is always the supremum from accept/reject sampling.

CHAPTER 2
BACKGROUND MATERIAL ON MONTE CARLO

2.1 Introduction

This chapter reviews the basics of Monte Carlo and Markov chain Monte Carlo required for the remainder of the dissertation. Recall from Chapter 1, interest lies in using Monte Carlo averages of simulated variables to estimate $\mu_h = \int_{\mathcal{X}_F} h(x)dF(x)$, where h is a Borel measurable function with finite variance with respect to F. We outline three methods, importance sampling, accept/reject sampling and the Metropolis Hastings algorithm, that are useful when direct simulation from F is difficult or impossible but direct simulation from another distribution similar to F is possible. We refer to the distribution similar to F as the *instrumental* distribution, and label it G. The three methods in this chapter modify G variates to obtain Monte Carlo approximations of expectations with respect to F. We note that when discussing accept/reject sampling or the Metropolis Hastings algorithm, G is often referred to as the *candidate* distribution and G variates as *candidates*. The title of the dissertation stems from the fact that our primary emphasis is the application of these two algorithms.

We begin with basic notation and definitions. Assume the distribution functions F and G are defined on $\mathbb{R}^k$ for some $k \geq 1$. We define a *support* of F to be an F measurable set, $\mathcal{X}_F$, such that $P_F(\mathcal{X}_F) = 1$ where P_F is the probability measure induced by F. Similarly, label a support of G by $\mathcal{X}_G$. The two basic assumptions for importance sampling (Section 2.3), accept/reject sampling (Section 2.4) and the independence Metropolis algorithm (Section 2.5), are

A1. F is absolutely continuous with respect to G;

A2. $C \equiv \text{ess. sup}\left\{\frac{dF}{dG}(x) \,|x \in \mathcal{X}_F\right\} < \infty$;

where the essential supremum in A2, defined as

$$C = \inf\left\{c : G\left(\left|\frac{dF}{dG}(x)\right| > c\right) = 0\right\},$$

is the supremum discarding sets of G measure 0. Technically, assumption A2 is not required for importance sampling or the independence Metropolis algorithm. However, both algorithms possess desirable properties when A2 holds. Furthermore, in the practical situations of interest for this dissertation, A2 will generally hold by the construction of G.

The absolute continuity assumption A1, ensures that $\frac{dF}{dG}$ exists (Billingsley; 1995, section 32, page 422). In most cases, F and G will both be absolutely continuous with respect to some product of Lesbegue and counting measures, say m. If $f = \frac{dF}{dm}$ and $g = \frac{dG}{dm}$ are the corresponding densities, then assumption A1 is verified if the support of g contains the support of f, in which case $C = \sup_{x \in \mathcal{X}_F} \frac{f(x)}{g(x)}$. Before we begin our discussion of Monte Carlo approximations we begin with an important approximation called the "Laplace approximation." The Laplace approximation is very useful for Monte Carlo as it may be used to construct accurate instrumental densities, g.

2.2 The Laplace Approximation

The Laplace approximation (Tierney et al.; 1989) is an analytic approximation to the expectations with respect to a distribution F with density f. We assume $\log f$ admits a Taylor expansion about the mode of f. Let $l(x) + a = \log f(x)$ for a constant a. Let $\hat{x}$ satisfy $l'(\hat{x}) = 0$, where $l'(x)$ is the vector of first derivatives of l evaluated at x. Similarly let $l''(x)$ be the matrix of second derivatives of l evaluated

at x. Then

$$\begin{aligned}\mu_h &= e^a \int h(x) \exp\left[l(x)\right] dx \\ &\approx e^a \int h(x) \exp\left[l(\hat{x}) + (x-\hat{x})^t l''(\hat{x})(x-\hat{x})/2\right] dx \qquad (2.1)\end{aligned}$$

where a superscript t represents the transpose of a matrix or vector. Setting $h(x) = 1$ we obtain

$$1 = \mu_1 \approx e^a \sqrt{2\pi} |l''(\hat{x})|^{-1/2} \exp\left[l(\hat{x})\right].$$

Therefore we may eliminate e^a by dividing (2.1) by this approximation of 1 yielding

$$\mu_h \approx \frac{1}{\sqrt{2\pi}} |l''(\hat{x})|^{1/2} \int h(x) \exp\left[(x-\hat{x})^t l''(\hat{x})(x-\hat{x})/2\right] dx. \qquad (2.2)$$

This suggests that f is approximately normal with mean $\hat{x}$ and variance $-(l''(\hat{x}))^{-1}$. Notice the interpretation in Bayesian settings is that the posterior is approximately normal with mean at the posterior mode and variance equal to the negative inverse of the curvature of the log posterior at that mode. In this dissertation we will not be interested in using the Laplace approximation as an approximation of μ_h. Rather, we use the Laplace approximation to estimate the mean and variance of F, and use these to construct an accurate instrumental density by shifting and scaling a heavy tailed t distribution by the approximate mean and standard deviation respectively.

2.3 Importance Sampling

Importance sampling weights i.i.d. G variates so that their weighted expectation corresponds to an unweighted expectation with respect to F. That is, importance sampling estimates μ_h with a weighted average of G variates. We denote expectation with respect to F and G by E_F and E_G respectively. Notice that if $\{X_i\}_{i\in\mathbb{N}}$ is an i.i.d. collection of G variates, then, under assumption A1 and

the strong law of large numbers,

$$\frac{1}{n}\sum_{i=1}^{n} h(X_i)\frac{dF}{dG}(X_i) \longrightarrow E_G\left[h(X_1)\frac{dF}{dG}(X_1)\right] = \mu_h, \tag{2.3}$$

with probability 1 as $n \to \infty$. This estimator is rarely used in practice as in nearly all applications of Monte Carlo $\frac{dF}{dG}$ is only known up to a constant of proportionality. To account for this, the estimator (2.3) is divided by

$$\frac{1}{n}\sum_{i=1}^{n} \frac{dF}{dG}(X_i) \longrightarrow E_G\left[\frac{dF}{dG}(X_1)\right] = 1,$$

to obtain the ratio estimator

$$\frac{\sum_{i=1}^{n} h(X_i)\frac{dF}{dG}(X_i)}{\sum_{i=1}^{n} \frac{dF}{dG}(X_i)} \longrightarrow \mu_h. \tag{2.4}$$

Notice that $\frac{dF}{dG}$ appears in both the numerator and denominator of the importance sampling estimator and hence it needs only to be known up to a constant of proportionality.

A Monte Carlo estimate is of little value without some manner of quantifying its Monte Carlo error. However, the asymptotic distribution of the importance sampling (ratio) estimator can be derived using the delta method (Sen and Singer; 1993, Theorem 3.4.5). Under moment conditions discussed later, the multivariate central limit theorem states that (Sen and Singer; 1993, Theorem 3.3.9)

$$\sqrt{n}\sum_{i=1}^{n}\left(\begin{bmatrix} h(X_i)\frac{dF}{dG}(X_i) \\ \frac{dF}{dG}(X_i) \end{bmatrix} - \begin{bmatrix} \mu_h \\ 1 \end{bmatrix}\right)$$

converges in law to a multivariate normal distribution with mean $\mathbf{0}$. Denote the variance of the asymptotic distribution by $\mathbf{\Sigma}$. If $q(x, y)$ is a real valued function then, using the delta method, we obtain that

$$\sqrt{n}\sigma^{-1}\left(q\left(\frac{\sum_{i=1}^{n} h(X_i)\frac{dF}{dG}(X_i)}{n}, \frac{\sum_{i=1}^{n} \frac{dF}{dG}(X_i)}{n}\right) - q(\mu_h, 1)\right) \tag{2.5}$$

converges in distribution to a standard normal distribution, where

$$\sigma^2 = q'(\mu_h, 1)^t \Sigma q'(\mu_h, 1).$$

By letting $q(a, b) = a/b$, we obtain the asymptotic distribution of the importance sampling estimate (2.4).

As we do not know σ^2, it must be estimated. By Slutsky's theorem, estimating σ^2 with a consistent estimator, such as the method of moments estimator, will not affect the asymptotic distribution of (2.5). Let $H_i = h(X_i)$ and $W_i = \frac{dF}{dG}(X_i)$. Then the method of moments estimator of σ^2 simplifies to

$$\hat{\sigma}^2 = n \left(\frac{\sum H_i W_i}{\sum W_i} \right)^2 \left(\frac{\sum H_i^2 W_i^2}{(\sum H_i W_i)^2} - 2 \frac{\sum H_i W_i^2}{(\sum H_i W_i)(\sum W_i)} + \frac{\sum W_i^2}{(\sum W_i)^2} \right). \tag{2.6}$$

Notice that this estimator does not depend on any constants of proportionality as W_i is always of the same order in both the numerator and denominator. Notice also that importance sampling only requires saving the sums $\sum H_i W_i$, $\sum W_i$, $\sum H_i^2 W_i^2$, $\sum H_i W_i^2$ and $\sum W_i^2$, and hence can be run by updating these sums rather than storing the simulated G variates.

The CLT holds and the variance estimate (2.6) is consistent provided

$$E_G[W_i^2] = E_G \left[\frac{dF}{dG}(X_i)^2 \right] < \infty$$

and

$$E_G[H_i^2 W_i^2] = E_G \left[h(X_i)^2 \frac{dF}{dG}(X_i)^2 \right] < \infty.$$

Thus, the assumption that the W_i are bounded almost surely with respect to G (assumption A2) is not strictly necessary. However, if A2 does hold, one need only verify that $E_G[H_i^2] < \infty$; i.e., h has finite G variance. This is generally not difficult in typical applications where G is a common distribution with well known properties.

2.4 Accept/Reject Sampling

Accept/reject sampling is a way to generate an i.i.d. sequence from F by thinning out an i.i.d. sequence from G. Our treatment of accept/reject sampling will be brief as it will be discussed in much further detail in Chapter 5. The accept/reject sampling algorithm is as follows.

Algorithm 2.1 Accept/Reject Sampling

1. Generate X from G and U a uniform$(0,1)$.
2. Accept X if $U \leq \frac{dF}{dG}(X)\frac{1}{C}$.
   Otherwise go to step 1.

We prove the validity of the accept/reject algorithm as the method of proof will be useful later. Note that this is a standard argument (see, for example, Robert and Casella; 1999, Lemma 2.3.1).

Theorem 2.1. *The random variables generated from Algorithm 2.1 are i.i.d. with distribution F.*

Proof. It is clear that the generated variables are independent and identically distributed. We show they follow distribution F. Let X be a G variate. Notice $\frac{dG}{dF}(X)\frac{1}{C} \leq 1$ with probability one by assumption A2. Thus

$$\begin{aligned}
P(X \leq x | X \text{accepted}) &= \frac{P\left(X \leq x, U \leq \frac{dF}{dG}(X)\frac{1}{C}\right)}{P\left(U \leq \frac{dF}{dG}(X)\frac{1}{C}\right)} \\
&= \frac{E_G\left[P\left(X \leq x, U \leq \frac{dF}{dG}(X)\frac{1}{C} | X\right)\right]}{E_G\left[P\left(U \leq \frac{dF}{dG}(X)\frac{1}{C} | X\right)\right]} \\
&= \frac{E_G\left[I_{(X \leq x)} P\left(U \leq \frac{dF}{dG}(X)\frac{1}{C} | X\right)\right]}{E_G\left[\frac{dF}{dG}(X)\frac{1}{C}\right]} \\
&= \frac{E_G\left[I_{(X \leq x)}\frac{dF}{dG}(X)\frac{1}{C}\right]}{\frac{1}{C}} \\
&= F(x)
\end{aligned}$$

(where, in the multivariate case, $\leq$ acts component-wise on the elements of x). □

Notice that the relation, $P(X\text{accepted}) = 1/C$, implies that the number of candidates required to obtain one target is geometric with success probability $1/C$.

2.5 The Metropolis Hastings Algorithm

The Metropolis Hastings algorithm (introduced to statisticians by Hastings; 1970) is a Markov chain Monte Carlo method (MCMC) of simulating a sequence of dependent random variables which has F as its stationary distribution. In this section we review the Metropolis Hastings algorithm in the form it is needed for the remainder of the dissertation. The material from this section was adapted from review articles by Tierney (1994), Chib and Greenberg (1995), Besag et al. (1995), Geyer (1992), and the book by Robert and Casella (1999).

The Metropolis Hastings (MH) algorithm generates candidates from another Markov chain, whose transition kernel we will denote by G, and thins them out in such a way that the resulting Markov chain has stationary distribution F. To be more specific, suppose that the Markov chain is at the current state x. The MH algorithm generates a candidate, y, from $G(y|x)$ and then flips a coin to decide whether the chain stays at x or moves to y. The success probability of the coin flip gives greater weight to whichever state is more consistent with F relative to its probability of being generated by G. For example, if a candidate has low G probability and high F probability, the chain will move to that point with a high probability, and then stay there for a long time.

In the special case where G does not depend on x the MH algorithm is referred to as the independence Metropolis algorithm (Tierney; 1994). With assumption A1, the chain moves to y with probability

$$\min\left(\frac{\frac{dF}{dG}(y)}{\frac{dF}{dG}(x)}, 1\right);$$

otherwise it stays at x. In this case assumption A2 implies the chain is *uniformly ergodic* (Robert and Casella; 1999, Theorem 6.3.1). That is, the distribution of the

chain converges in total variation to F at a geometric rate, where this rate does not depend on the initial state of the chain. The converse to this theorem says that if A2 does not hold then the rate is sub-geometric.

In the general Metropolis Hastings algorithm, G depends on the current state x. For simplicity, assume F has density f and $G(y|x)$ corresponds to a conditional density $g(y|x)$, which is referred to as the candidate transition density. Then, the Metropolis Hastings algorithm moves to y with probability

$$\min\left(\frac{f(y)g(x|y)}{f(x)g(y|x)}, 1\right); \tag{2.7}$$

otherwise the chain stays at x. Provided the resulting chain is irreducible and aperiodic, the Markov chain will converge in total variation to F, and averages from the Markov chain will converge to the associated mean of the stationary distribution. Robert and Casella (1999) give this result as well as conditions on g which ensure that the Metropolis algorithm will be irreducible and aperiodic.

Our applications of the Metropolis Hastings algorithm require a little modification. In particular, we are concerned with situations where g is chosen at random from a class of candidate transition densities indexed by a random variable θ. We label $g = g(y|x,\theta)$ and assume the index variable θ is generated from $\pi(\theta)$ independently of x. We denote the support of π by $\mathcal{X}_\pi$.

Consider a MH chain run with θ fixed. The acceptance probability of a candidate y would be

$$\rho(y|x,\theta) \equiv \min\left(\frac{f(y)g(x|y,\theta)}{f(x)g(y|x,\theta)}, 1\right).$$

Allow $k(y|x,\theta)$ to be the resulting transition density from this chain. We are interested in settings where (for θ fixed) $k(y|x,\theta)$ does not produce an irreducible chain. We note that even if $k(y|x,\theta)$ does not produce an irreducible chain, it still

satisfies *detailed balance*. That is, $k(y|x,\theta)$ satisfies $f(x)k(y|x;\theta) = f(y)k(x|y;\theta)$ (see the proof of Theorem 6.2.3 in Robert and Casella; 1999).

Consider now a Markov Chain that is run by first picking a θ and then updating the chain with $k(y|x,\theta)$. We outline this algorithm below. Assume the chain is at initial state x. We are interested in the chain marginalized over the

Algorithm 2.2 Random Index Metropolis/Hastings Algorithm

1. Generate θ from π
2. Generate y from $g(y|x,\theta)$
3. Generate u a uniform$(0,1)$
4. If $u \leq \rho(y|x,\theta)$ move the chain to y. Otherwise chain stays at x.

index variable θ. This chain has transition density

$$k(y|x) = \int_{\mathcal{X}_\pi} k(y|x;\theta)d\pi(\theta).$$

Note that $k(y|x)$ satisfies the detailed balance equation,

$$\begin{aligned} f(x)k(y|x) &= \int_{\mathcal{X}_\pi} f(x)k(y|x;\theta)d\pi(\theta) \\ &= \int_{\mathcal{X}_\pi} f(y)k(x|y;\theta)d\pi(\theta) \\ &= f(y)k(x|y). \end{aligned}$$

Hence, provided it is irreducible and aperiodic, the resulting chain will have F as its stationary density (Robert and Casella; 1999, Theorem 6.2.2). In Chapter 4, these conditions will be established for our application of this Markov chain.

We now show that some common Markov chain algorithms fall under this setting. We focus on particular Markov chains discussed in applications later. First, consider the case where $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$ for $k > 1$. Let $\pi(\theta)$ be the discrete uniform distribution on the integers $1, \ldots, k$. Let

$$g(\mathbf{y}|\mathbf{x};\theta) \propto f(x_1, \ldots, x_{\theta-1}, y_\theta, x_{\theta+1}, \ldots, x_k).$$

Each $g(\mathbf{y}|\mathbf{x};\theta)$ is called a full conditional of f. Note that $\min\left(\frac{f(\mathbf{y})g(\mathbf{x}|\mathbf{y},\theta)}{f(\mathbf{x})g(\mathbf{y}|\mathbf{x},\theta)}, 1\right) = 1$, and hence the Metropolis algorithm has no rejection. This algorithm is referred to as the "random scan Gibbs sampler". In Section 3.6 we will see how random scan Gibbs sampling applies to exact Monte Carlo analysis of contingency tables. Other versions of the (random scan) Gibbs sampler, such as block Gibbs and hybrid Gibbs samplers, are easily described as modifications of this algorithm.

As a second example, for an arbitrary function $v : \mathcal{X}_F \times \mathcal{X}_\pi \to \mathcal{X}_F$, let

$$g(y|x;\theta) = \begin{cases} 1 & \text{for } y = v(x,\theta) \\ 0 & \text{otherwise} \end{cases}.$$

That is the candidate entirely depends on θ through v. For example we might let $y = v(x,\theta) = x + \theta$ or $y = v(x,\theta) = x\theta$. The Metropolis Hastings acceptance probability then takes the form $\min\left(\frac{f(y)}{f(x)}, 1\right)$. When $v(x,\theta) = x + \theta$ this algorithm is referred to as the random walk Metropolis algorithm. In Section 3.7 we will see a random walk Metropolis algorithm also applies to exact analysis of contingency tables. In that example θ is chosen at random from a generating set for $\mathcal{X}_F$.

We end this section with a brief discussion of convergence control for Markov chain Monte Carlo algorithms. Ideally, if $\{Y_i\}$ is the sequence of states of the Markov chain, we would like the average $\frac{1}{n}\sum_{i=1}^{n} h(Y_i)$ to converge almost surely to μ_h, and, properly normalized, to converge in law to a standard normal distribution. Our discussion of these matters will be greatly aided in that all of our applications of MCMC utilize reversible, aperiodic, irreducible chains. Under such conditions and the assumption,

$$\gamma_h^2 = \sum_{k=-\infty}^{\infty} \mathrm{Cov}_F[h(Y_0)h(Y_k)] < \infty,$$

then

$$\frac{1}{n}\sum_{i=1}^{n} h(Y_i) \longrightarrow \mu_h,$$

and

$$P\left(\sqrt{n}\frac{\sum_{i=1}^{n}(h(y_i)-\mu_h)}{\gamma_h}\leq y\right)\longrightarrow\Phi(y) \tag{2.8}$$

(Robert and Casella; 1999, Theorems 4.7.4 and 4.7.7). Further, (Geyer; 1992, Theorem 2.1) claims that, under the assumptions above, $\mathrm{Var}(\sum_{i=1}^{n}h(Y_i))$ converges to γ_h^2. Consistently estimating γ_h is a difficult problem in Markov chain Monte Carlo. The naive estimate obtained by adding up the estimates of $\mathrm{Cov}_F[h(Y_0)h(Y_l)]$, for $l=0,\ldots,n-1$, is known to be inconsistent in general (Geyer; 1992). Down-weighting the larger lag autocovariances using a "window" may produce a consistent estimate (this is the method suggested by Hastings in his seminal paper). However, this approach has two drawbacks. First, consistency will be dependent on the choice of window and second, estimating the largest lag autocovariances requires storing the entire chain. Despite these drawbacks we note that estimating the autocovariances of the Markov chain is valuable whether or not the estimates are used to approximate γ_h.

Technically, the strong law of larger numbers (referred to as the *ergodic* theorem) and the central limit theorem (2.8), are proved under the assumption the chain is at stationarity. The assumption that the chain is at stationarity is not as restrictive as it may seem. Geyer (1992) points out that, provided the chain is Harris recurrent, the convergence properties of the chain do not depend on the starting point. In our applications the Markov chain will reside on a finite state space, where Harris recurrence is simply positive recurrence.

An alternative approach to convergence control uses regeneration times. Assume the state space of our Markov chain is finite and let $k(y|x)$ be the transition probability of going from state x to state y. Fix a state of the Markov Chain, $x_r\in\mathcal{X}_F$. Then, tours between returns to x_r are independent. Therefore, provided the chain hits x_r frequently enough, averages between returns to x_r are independent and standard theorems may be used to estimate the Monte Carlo error (see, for

example Mykland et al.; 1995). When the chain hits x_r it is said to *regenerate*. This approach is not feasible when the state space is too large and returns to x_r are too infrequent.

It would be more useful if a regeneration could occur whenever the chain hits a set rather than a specific state. This is the case if $k(y|x)$ is constant on some set A, that is $k(y|x) = p(y)$ when $x \in A$. Then a regeneration occurs whenever the chain hits the set A. Such a set is called an *atom* (clearly, by the previous paragraph, a single state in a finite state space chain is an atom). As transition densities possessing this property are rare in practice, suppose instead k satisfies: $k(y|x) \geq \epsilon p(y)$ for some mass function p and $x \in A$. Then, for $x \in A$

$$k(y|x) = \epsilon p(y) + (1 - \epsilon)\frac{k(y|x) - \epsilon p(y)}{1 - \epsilon}.$$

Then, if the previous state was in the set A, a coin flip would determine if a regeneration occurred. Of course, there is an inverse relationship between the sizes of A and ϵ depending on tightness of the inequality $k(y|z) \geq \epsilon p(y)$. In Chapter 4 we discuss the possibility of using regeneration times for convergence control for our applications of MCMC accounting for this issue.

The last method of convergence control we discuss for our problem is batching. The batch means method, in its simplest form, treats non-overlapping batches of equal size of the Markov chain as if they were independent. To use this approach, the Monte Carlo practitioner must have a feel for the dependence in the chain, suggesting a study of the autocovariances. Batch means is a simple (albeit ad-hoc) method for quantifying the Monte Carlo error in Markov chains. In Chapter 4 we empirically study the performance of the batch means method for our application of MCMC.

2.6 Discussion

We end this chapter with a discussion of how the independence Metropolis algorithm may be used in conjunction with rejection to eliminate the need to know the exact supremum C (Tierney; 1994). Suppose a rejection sampler is run replacing C with a lower bound, say T. Then is is easy to see that the accepted variates have density $\tilde{f} \propto \min(f, gT)$. Tierney (1994) suggests using $\tilde{f}$ as the candidate density for an independent Metropolis Algorithm. By plugging $\tilde{f}$ into the Metropolis acceptance probability, we obtain

$$\rho(y|x) = \min\left(\frac{f(y)\tilde{f}(x)}{f(x)\tilde{f}(y)}, 1\right) = \begin{cases} 1 & \text{if } f(x) \leq Tg(x) \\ \frac{g(x)T}{f(x)} & \text{if } f(y) \leq Tg(y) \\ & \text{and } f(x) > Tg(x) \\ \min\left(\frac{f(y)g(x)}{f(x)g(y)}, 1\right) & \text{otherwise} \end{cases}.$$

As before, let $k(y|x)$ be the transition density of the resulting Markov chain. Note that $k(y|x) = \tilde{f}(y)$ for $x \in A \equiv \{x|f(x) \leq Tg(x)\}$. That is, for $x \in A$ the Metropolis Hastings algorithm always moves to y. That is, the set A is an atom and returns to A are regenerations.

CHAPTER 3
EXACT CONDITIONAL ANALYSIS

3.1 Nuisance Parameters

In this Chapter we review exact conditional analysis, an inferential method carried out after eliminating nuisance parameters by conditioning on their sufficient statistics. We begin with an overview of nuisance parameters and the methods for dealing with them. Basu (1977) gives an excellent review of this topic, including the Bayesian approach which is not covered in this dissertation. To set up the problem, assume the response, $\mathbf{Y} = (Y_1, \ldots, Y_n)'$, follows the distribution $F(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda})$ and interest lies in testing

$$H_0 : \boldsymbol{\lambda} = \boldsymbol{\lambda}_0 \tag{3.1}$$

against an alternative, which we leave unspecified. We adopt the convention throughout of writing random variables in upper case and realized values, such as the arguments of a density, in lower case. Further, vectors will always be bold type and we label the observed data as $\mathbf{y}_{obs}$.

Tests based on the null hypothesis (3.1) are useful, for example, to test the nature of $\boldsymbol{\lambda}$'s influence on the response, to compare the fit of the simpler model $F(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda}_0)$, to invert to obtain a confidence set for $\boldsymbol{\lambda}$, and so on. We refer to $F(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda}_0)$ as the *null distribution* and $F(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda})$ as the *alternative distribution*. Notice that, exact probability calculations under the null hypothesis require accounting for the unknown $\boldsymbol{\beta}$. As $\boldsymbol{\beta}$ is not of interest for the test under consideration, it is called a *nuisance parameter*. Throughout this chapter all nuisance parameters will be labeled some version of β and all parameters specified under the null hypothesis as some version of λ.

Table 3.1: A 2×2 table.

Treatment	Successes	Failures	Total
A	y_1	$n_1 - y_1$	n_1
B	y_2	$n_2 - y_2$	n_2

Without a doubt, the most famous example of a nuisance parameter is the two sample binomial problem. Let Y_1 and Y_2 be binomial counts with log odds of success β and $\beta + \lambda$ and number of trials n_1 and n_2 respectively. For example Y_1 and Y_2 might be success counts in a two treatment clinical trial. The data is usually summarized in a 2×2 table as in Table 3.1. The null hypothesis (3.1) with $\lambda_0 = 0$ implies that the odds of success is constant for the two treatments. Note that any probability calculations under the null hypothesis require the elimination of β. In the rest of this section we outline methods for handling nuisance parameters in a general framework. Before continuing, it should be noted that the 2×2 table has a long history, dating back nearly a century. Excellent discussions regarding analysis of the 2×2 table are given by Yates (1984) and Upton (1982) who present alternative views on the problem. More recent accounts are given by Greenland (1991), and Routledge (1992).

Let h be a test statistic of interest with observed value h_{obs}, that is $h(\mathbf{y}_{obs}) = h_{obs}$. Ideally, the distribution of $h(\mathbf{y})$ will be ancillary for $\boldsymbol{\beta}$ (the distribution of h does not depend on $\boldsymbol{\beta}$). For example, let $\mathbf{Y}_1 = (Y_{11}, \ldots, Y_{1n})'$ and $\mathbf{Y}_2 = (Y_{21}, \ldots, Y_{2n})'$ be mutually independent following the location scale density $\beta_1^{-1} f(\frac{y_{ij} - \mu_i}{\beta_1})$, with $\mu_1 = \beta_2$ and $\mu_2 = \beta_2 + \lambda$. Under the null hypothesis of equal locations, $H_0 : \lambda = 0$, the statistic

$$\frac{\bar{Y}_1 - \bar{Y}_2}{\sqrt{SS_Y}}, \tag{3.2}$$

where $SS_Y = \sum_{i=1}^{2} \sum_{j=1}^{n} (Y_{ij} - \bar{Y}_i)^2$, is ancillary to the nuisance parameter $\boldsymbol{\beta} = (\beta_1, \beta_2)'$. Therefore, if (3.2) is the test statistic of interest, $\boldsymbol{\beta}$ need not be considered.

Another example where an exact pivot is available was suggested by Anthony Davison (personal communication). As this test will be useful later, we briefly outline Davison's argument. Consider a score test of $H_0 : \lambda = 0$ versus $H_a : \lambda > 0$ for i.i.d. observations, $Y_1, \ldots Y_n$, from a generalized Pareto distribution. The generalized Pareto distribution is specified by a Taylor expansion of the log of the density,

$$\log f(y; \beta, \lambda) = -\log\beta - \frac{y}{\beta} + \lambda\left(\frac{y^2}{\beta^2} - \frac{y}{\beta}\right) + \lambda^2\left(\frac{y^3}{\beta^3} + \frac{y^2}{\beta^2}\right) + \ldots,$$

where $\lambda \geq 0$. Notice that when $\lambda = 0$ the generalized Pareto distribution is the exponential distribution with mean β. Therefore, under the null hypothesis, $\hat{\beta} = \bar{y}$ (where $\hat{\beta}$ is the ml estimate of β). Further note that $\frac{\partial \log f}{\partial \lambda}\big|_{\lambda=0} = \frac{y^2}{\beta^2} - \frac{y}{\beta}$ and hence the score statistic is equivalent to $h = \sum \frac{y_i^2}{(\sum y_i)^2}$. This statistic is ancillary to β. In fact, it is easy to show that under H_0, h follows the distribution of the sum of the squared spacings of a sample of $n-1$ uniform random variables. This statistic is called Greenwood's statistic (see, for example, Pyke; 1965) and is used as a diagnostic in Chapter 5.

Another option to eliminate $\boldsymbol{\beta}$ is to condition on its sufficient statistic. We introduce conditional analysis as we will need it for the dissertation. A more complete treatment can be found in the review article by Reid (1995). Let $\mathbf{S}$ be a minimal sufficient statistic for $\boldsymbol{\beta}$ under the null hypothesis. We assume $\mathbf{S}$ is not the data itself, because, if there is no reduction in the data through $\mathbf{S}$, then the conditional distribution of $\mathbf{y}$ given $\mathbf{S}$ is degenerate and therefore not useful for inference. Assuming this is not the case, then $F(\mathbf{y}|\mathbf{s}, \boldsymbol{\lambda}_0)$ does not depend on $\boldsymbol{\beta}$, and may be used for inference about $\boldsymbol{\lambda}_0$. For example, let $\mathbf{s}_{obs}$ be the observed sufficient statistic, then the conditional P-value, $P(h(\mathbf{Y}) \geq h_{obs}|\mathbf{S} = \mathbf{s}_{obs})$, can be used to assess the validity of the null hypothesis or to perform a fixed α-level test. When there is no confusion, we write the conditional P-value as $P(h \geq h_{obs}|\mathbf{S})$.

If the goal is to use the conditional P-value as a descriptive measure it has the nice interpretation of being the uniform minimum variance unbiased estimator of $P(h \geq h_{obs}; \boldsymbol{\beta})$ provided $\mathbf{S}$ is complete. That is, the conditional P-value is the best estimator of the P-value we would use if $\boldsymbol{\beta}$ were known. If, instead, the conditional P-value is used for a fixed α-level test, it is justified by the fact that the nominal α-level type I error bound is preserved unconditionally (a fact that is proved later in this section). It is from this property that conditional tests enjoy the qualification "exact".

The "exactness" of conditional tests (as well as other exact tests) can be viewed in a positive or negative way. By definition, exact tests control for values of the nuisance parameter that may be unrealistic or unsupported by the data. In discrete settings, the conservativeness of exact conditional tests can often be quite extreme. For example, the conditional distribution for binary logistic regression where one of the nuisance parameters is a regression parameter for a continuous covariate is often degenerate at the observed data. Then, regardless of the test statistic, the conditional P-value is 1. In this situation, and others where the conditional distribution is extremely discrete, conditional analysis is of little value.

Another criticism of conditional tests is that the distribution of $\mathbf{S}$ will usually depend on $\boldsymbol{\lambda}$ under the alternative hypothesis. For example, in the two sample binomial problem from Table 3.1, the sufficient statistic for β is $S = Y_1 + Y_2$. Under the alternative hypothesis S is the sum of binomials with success probabilities $(1 + \exp(-\beta))^{-1}$ and $(1 + \exp(-\beta - \lambda))^{-1}$ respectively; and thus its distribution depends on λ. As λ is the parameter of interest critics of conditional tests claim conditioning on S removes relevant information. A final criticism of conditional tests is that they can be very difficult to perform. This problem, with regard to log-linear models, is directly addressed in Chapter 4.

Despite these criticisms, we adopt the position that conditional tests are sufficiently justified to merit their study. First, conservativeness can be justified in many situations where the experimenter must be absolutely sure the type one error rate is adhered to. Secondly, in the situations of interest for this dissertation, log-linear models for contingency tables, the conditional distributions are not overly discrete (Agresti; 1992) and hence the conclusions are not overly conservative.

As a final justification we note that conditional tests can be thought of as a generalization of the standard approach in most normal theory tests. To illustrate, in the previous example of testing equality of locations from two independent location/scale populations, we claimed the t statistic (3.2) is ancillary to the common location, β_1, and the common scale, β_2. If the location/scale density is the normal density, then the minimal sufficient statistic for $\boldsymbol{\beta} = (\beta_1, \beta_2)'$ is also complete (arguments for this may be found in Casella and Berger; 1990), and hence independent of (3.2) by Basu's theorem. Therefore, one can think of the t test as a conditional test, the conditioning is just irrelevant. This argument generalizes to nearly all standard normal theory tests.

One class of exact alternatives to conditional tests are called *unconditional* tests. The attained significance level for the unconditional test is the largest possible P-value with known $\boldsymbol{\beta}$. That is, the unconditional P-value is

$$\sup_{\boldsymbol{\beta}} P(h \geq h_{obs}; \boldsymbol{\beta}).$$

As this quantity is larger than $P(h \geq h_{obs}; \boldsymbol{\beta})$ for any value of $\boldsymbol{\beta}$, this approach produces an exact test. Barnard (1945) initially introduced this P-value for the 2×2 table. The arguments comparing conditional and unconditional tests still endure. The previously mentioned review article by Yates (1984) is the best source on this topic (in support of the conditional approach), while Suissa and Shuster (1984) and Berger and Boos (1994) support unconditional approaches.

Interestingly, we note that unconditional tests and conditional tests can be embedded in a class of exact tests based on approximate conditioning. Let $D(\mathbf{s}, \mathbf{s}_{obs})$ be a metric measuring the distance between $\mathbf{s}$ and $\mathbf{s}_{obs}$. Then consider the P-value

$$p_d \equiv \sup_{\boldsymbol{\beta}} P(h(\mathbf{Y}) \geq h_{obs} | \{D(\mathbf{S}, \mathbf{s}_{obs}) \leq d\}, \boldsymbol{\beta}). \tag{3.3}$$

The conditional P-value corresponds to $d = 0$ and the unconditional P-value corresponds to $d \geq \max(D(\mathbf{s}, \mathbf{s}_{obs}))$. As p_d is a function of the observed data, $\mathbf{y}_{obs}$ we write $p_d(\mathbf{y}_{obs})$.

We now argue that use of p_d results in an exact test. For a fixed α and d, this P-value induces a rejection region,

$$RR = \{\mathbf{z} \text{ in the sample space } | p_d(\mathbf{z}) \leq \alpha\}.$$

Let $\mathbf{z}$, with sufficient statistic $\mathbf{s}_{\mathbf{z}}$, be an element of RR with a minimum value of h. Then

$$\begin{aligned} P[\mathbf{Y} \in RR | \{D(\mathbf{S}, \mathbf{s}_{\mathbf{z}}) \leq d\}] &= P[h(\mathbf{Y}) \geq h(\mathbf{z}) | \{D(\mathbf{S}, \mathbf{s}_{\mathbf{z}}) \leq d\}] \\ &\leq p_d(\mathbf{z}) \\ &\leq \alpha \end{aligned}$$

Taking expectations yields $P[\mathbf{Y} \in RR; \boldsymbol{\beta}] \leq \alpha$. That is, the probability of a type I error is less than α for any value of $\boldsymbol{\beta}$. Note that this argument holds for $d = 0$ and hence we have proven that the conditional P-value produces an unconditional level α test. Such approximate conditional approaches have been considered, for example in Cox and Reid (1987) and Pierce and Peters (1999), who use a profile likelihood rather than taking the supremum over $\boldsymbol{\beta}$. One possibility that has not been explored is to take $\inf_d p_d(\mathbf{y}_{obs})$ as the attained significance level.

We continue with our study of nuisance parameters noting that the most common way to eliminate $\boldsymbol{\beta}$ in problems where no exact pivot is available is to

use the fact that most test statistics of interest have a limiting distribution under the null hypothesis that does not depend on $\boldsymbol{\beta}$. For example, for contingency tables, the class of power divergence statistics (Read and Cressie; 1988) which include the deviance and the Pearson chi-squared statistic, have limiting chi-squared distributions. This approach has the benefit of being easy to implement. However, since the sample is necessarily finite, the true type I error rate may differ drastically from the nominal value. For example, using chi-squared tests for sparse contingency tables can yield extremely conservative results that totally disagree with the results of exact tests that guarantee the nominal type one error rate (compare the exact conditional and asymptotic P-values for the pathologist data in entry 4 of Table 4.1).

Another alternative is to use the parametric bootstrap. To illustrate, let h_{obs} be the observed value of our test statistic h. A bootstrap P-value is estimated by simulating pseudo datasets or resamples, $\mathbf{y}^*$, from $F(\mathbf{y}; \hat{\boldsymbol{\beta}}, \boldsymbol{\lambda}_0)$ and calculating the proportion of times $h(\mathbf{y}^*)$ is larger than h_{obs}. Note that this may be computationally intensive as generally h will depend on the fitted values corresponding to each simulated $\mathbf{y}$. Note also that the bootstrap P-value is itself a statistic and so one may iterate the bootstrap procedure, referred to as the *double bootstrap*. Obviously, one can re-apply the bootstrap arbitrarily many times. Presnell (1996) applies this method to McNemar's test for comparing dependent proportions. Amazingly, the infinitely iterated bootstrap can be calculated without resampling in this context. His procedure could be extended to the situations covered in this dissertation, but, resampling would be necessary. Although the use of the plug-in method and the iterated bootstrap does not guarantee the type I error rates, they do center the null distribution around a value of $\boldsymbol{\beta}$ supported by the data. In contrast, conservative exact tests that guarantee the type I error rate must consider values of $\boldsymbol{\beta}$ that are unrealistic given the data.

3.2 Log-linear Models

In this section we outline the various concepts necessary for performing conditional tests for log-linear models. For simplicity only Poisson log-linear models are considered although we note that multinomial models are included as special cases. Specifically, any multinomial log-linear model can be nested within a Poisson log-linear model in such a way that the reference set and the probabilities defined on it, obtained by conditioning on sufficient statistics, will be the same for the two sampling assumptions. We illustrate this with an example at the end of this section. Detailed reviews and discussion of conditional inference for categorical data can be found in Agresti (1992) and Agresti (2001).

Suppose that data, $\mathbf{Y} = (Y_1, \ldots, Y_n)'$, are independent Poisson random variables with means $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)'$. Further assume that the means satisfy

$$\log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\lambda}, \tag{3.4}$$

where $\begin{bmatrix} \mathbf{X} & \mathbf{Z} \end{bmatrix}$ is a full rank $n \times w$ matrix ($w \leq n$) and $\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\lambda} \end{bmatrix}$ is a w-vector of unknown regression parameters. The model is called *saturated* when $w = n$. Frequently it is desired to compare model (3.4) with a simpler model,

$$\log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\beta}, \tag{3.5}$$

by, testing $H_o : \boldsymbol{\lambda} = 0$. Model (3.4) is referred to as the alternative model while model (3.5) is referred to as the null or restricted model. The components of $\boldsymbol{\beta}$ are nuisance parameters for this test. Let $f(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda} = 0)$ denote the joint mass function of the data, under the null model given by

$$f(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda} = 0) = \prod_{i=1}^{n} \frac{\mu_i^{y_i} \exp(-\mu_i)}{y_i!} = \frac{1}{\prod_{i=1}^{n} y_i!} \exp(\boldsymbol{\beta}'\mathbf{X}'\mathbf{y} - \mu_+) \tag{3.6}$$

where $\mu_+ = \sum_{i=1}^n \mu_i$. Observe from (3.6) that the minimal sufficient statistic for $\boldsymbol{\beta}$ is $\mathbf{S} = \mathbf{X}'\mathbf{Y}$, with null mass function, $f(\mathbf{s}; \boldsymbol{\beta}, \boldsymbol{\lambda} = 0)$, given by

$$f(\mathbf{s}; \boldsymbol{\beta}, \boldsymbol{\lambda} = 0) = \sum_{\mathbf{X}'\mathbf{y}=\mathbf{s}} f(\mathbf{y}; \boldsymbol{\beta}, \boldsymbol{\lambda} = 0) = \left(\sum_{\mathbf{X}'\mathbf{y}=\mathbf{s}} \frac{1}{\prod_{i=1}^n y_i!} \right) \exp(\boldsymbol{\beta}'\mathbf{s} - \mu_+). \tag{3.7}$$

Combining (3.6) and (3.7) gives the conditional mass function

$$f(\mathbf{y}|\mathbf{s}; \boldsymbol{\lambda} = 0) = \left(\sum_{\mathbf{X}'\mathbf{y}=\mathbf{s}} \frac{1}{\prod_{i=1}^n y_i!} \right)^{-1} \frac{1}{\prod_{i=1}^n y_i!}. \tag{3.8}$$

We write $f(\mathbf{y}|\mathbf{s})$ for (3.8) when there can be no confusion over the value of $\boldsymbol{\lambda}$. The support of this mass function at $\mathbf{s} = \mathbf{s}_{obs}$, is called the reference set and is generally labeled $\Gamma = \{\mathbf{y} \in \mathbb{Z}_+^n | \mathbf{X}'\mathbf{y} = \mathbf{s}_{obs}\}$, (where $\mathbb{Z}_+$ denotes the non-negative integers).

As an example consider $I \times J$ contingency tables. The saturated model is usually specified as

$$\log(\mu_{ij}) = \beta + \beta_i^1 + \beta_j^2 + \lambda_{ij}.$$

where the last levels of each parameter are set to 0 for identifiability (note that the superscripts on the "beta" parameters are identifiers and not powers). The independence model corresponds to $\lambda_{ij} = 0$ for all i and j. Testing the independence model versus the saturated model corresponds to testing $H_o : \lambda_{ij} = 0 \; \forall \; i, \; j$. The sufficient statistics for the nuisance parameters, β_i^1 and β_j^2, are the row and column totals, hence the reference set is all $I \times J$ tables with the same margins as the observed data.

We now illustrate how multinomial sampling can induce the same conditional inferences as independent Poisson sampling, using the independence model as an example. In the notation of the previous paragraph, the sufficient statistic for β is $\sum_{ij} y_{ij}$. It is then easy to see that the distribution of $\mathbf{Y}|\sum_{ij} y_{ij}$ under the null hypothesis is multinomial with i, j cell probabilities proportional to $e^{\beta_i^1} e^{\beta_j^2}$. That is, the cell probabilities factor into a row probability and a column probability. This

is a multinomial model that specifies independence between rows and columns. To summarize, the null conditional distribution for any product multinomial log-linear model is exactly the same as the null conditional distribution for the corresponding Poisson log-linear model. This result is similar to the classical result that corresponding Poisson and product multinomial log-linear models produce the same maximum likelihood estimates and power-divergence statistics (see Fienberg; 2000).

Recall that larger values of h lend support to the alternative hypothesis. The conditional P-value is defined as:

$$\begin{aligned} P &= P\left(h \geq h_{obs} | \mathbf{S} = \mathbf{s}_{obs}\right) \\ &= \sum_{\mathbf{y} \in \Gamma} I\{h(\mathbf{y}) \geq h_{obs}\} f(\mathbf{y}|\mathbf{s}_{obs}) \\ &= \frac{\sum_{\mathbf{y} \in \Gamma} I\{h(\mathbf{y}) \geq h_{obs}\} (\prod_{i=1}^{n} y_i!)^{-1}}{\sum_{\mathbf{y} \in \Gamma} (\prod_{i=1}^{n} y_i!)^{-1}}, \end{aligned} \tag{3.9}$$

where $I\{\}$ denotes an indicator function.

We note that it is preferable for h to be an unconditionally constructed statistic. For example if h is a likelihood ratio test, it is preferable for h to be formed as

$$\frac{\sup_{\boldsymbol{\beta}} f(\mathbf{y}; \boldsymbol{\beta}; \boldsymbol{\lambda} = 0)}{\sup_{\boldsymbol{\beta}, \boldsymbol{\lambda}} f(\mathbf{y}; \boldsymbol{\beta}; \boldsymbol{\lambda})},$$

rather than

$$\frac{f(\mathbf{y}|\mathbf{s}; \boldsymbol{\lambda} = 0)}{\sup_{\boldsymbol{\lambda}} f(\mathbf{y}|\mathbf{s}; \boldsymbol{\lambda})}.$$

The two constructions are equal when $\mathbf{S}$ is ancillary for $\boldsymbol{\lambda}$. The reason the unconditional construction is preferred is that the conditioning is done to produce an unconditional level α test and therefore h should have an unconditional interpretation. This argument does not hold in cases, such as item response models, where the conditional distribution is the distribution of interest and the conditioning is not just a tool to eliminate nuisance parameters.

3.3 Enumeration

Exact computation of the conditional P-value (3.9) requires enumeration of the reference set, Γ. This task has been well developed for the independence model (Boulton and Wallace; 1973; Mehta and Patel; 1983). Papers by Pagano and Tritchler (1983), Hirji (1996), Hirji et al. (1987), and Yao and Tritchler (1993) present approaches for other specific models. The approach of Pagano and Tritchler (1983) applies to Zelen's test for $2 \times 2 \times k$ tables (Zelen; 1971) by inverting the characteristic function of the conditional mass function with the fast Fourier transform.

The method that receives the most attention is the *network algorithm* (Mehta and Patel; 1980). In theory, the network algorithm may be applied very generally. This algorithm stores the reference set as a network of recursively generated nodes and arcs. A backwards sweep through the network throws out terminal nodes (terminal nodes are nodes that do not have outgoing arcs). The resulting network is equivalent to Γ where one path through the network represents one point in Γ. The network algorithm has been implemented for a variety of algorithms including the independence model (Mehta and Patel; 1983) and exact logistic regression (Mehta et al.; 2000).

Recent work by Casella and Wells (work in progress) shows how to represent the reference set as the solution set to a system of polynomial equations. We illustrate their method for the 2×2 table with notation given in Table 3.1. Denote the observed sufficient statistic s_{obs} and recall that $0 \leq y_1 \leq n_1$ and $0 \leq y_2 \leq n_2$. The reference set consist of those points, y_1, y_2, that are zeros of the system of

polynomials

$$\begin{aligned} y_1 + y_2 - s_{obs} &= 0, \\ \prod_{i=0}^{n_1}(y_1 - i) &= 0, \\ \prod_{i=0}^{n_2}(y_2 - i) &= 0. \end{aligned}$$

Casella and Wells show how to represent the solutions to this system of equations by a sequence of polynomials structured so that the first polynomial involves only y_1 and the second only y_1 and y_2 (in a similar manner to Gauss Jordan elimination for linear equations). This collection of polynomials is called a Groebner basis. For example, for this problem, the reference set is easily calculated to be

$$\{y_1 \in \mathbb{Z}_+ | \max(s - n_2, 0) \leq y_1 \leq \min(n_1, s)\}$$

and $y_2 = s - y_1$. Therefore the reference set is the zeros (in $\mathbb{Q}$) of

$$\begin{aligned} p_1 &= \prod_{i=\max(s-n_2,0)}^{\min(n_1,s)} (y_1 - i) \\ p_2 &= y_1 + y_2 - s. \end{aligned}$$

Technically, we have not shown that p_1, p_2 forms a Groebner basis. To do this we use Theorem 6 on page 82 of Cox et al. (1997). In their notation, it is easy to show $S(p_1, p_2) = p_1 + y_1^{a-1} p_2$ where $a = \max(s - n_2, 0) - \min(n_1, s)$[1] . This is a sufficient condition for p_1, p_2 to be a Groebner basis.

In general, for this approach to be meaningful, the number of operations required in calculating the Groebner basis and finding its zeros must be less than $(n_1 + 1)(n_2 + 1)$, the number of operations to calculate the reference set in the most obtuse way, using two nested "for" loops. This argument extends to exact

[1] Under lexicographic ordering with $y_1 > y_2$.

logistic regression, where the Groebner basis approach will only be useful (as an enumeration tool) if the total number of operations is less than the product of one plus the sample sizes. Later on, we will show two ways in which Groebner bases are useful for Monte Carlo approximations.

Overall, the feasibility of enumeration largely relies on the size or complexity of Γ. It is easy to construct examples where the size of Γ is too large to enumerate even if a perfect algorithm was available that only cycled through every member of Γ once. For example, the reference set for the independence model for the 5×5 table from Chapter 1 possesses over twelve billion tables. The reference set for the uniform association model only possesses 34,666 tables. However, efficiently cycling through these 34,666 tables without over counting is difficult. Clearly an enumerate and reject strategy of enumerating the 12 billion tables in the independence model to obtain the 34 thousand in the uniform association model is unsatisfactory. Calculating the Groebner basis for that problem is also infeasible, as it involves working with polynomials of very high degrees. The difficulty in extending the network algorithm to models like uniform association lies in establishing *termination rules* (Mehta et al.; 2000). These rules determine which nodes are terminal during the recursive generation of the network. The tighter the criteria for determining terminal nodes, the more efficient the network algorithm is. Determining these rules is a non-trivial task, as illustrated in Hirji et al. (1987).

3.4 Monte Carlo Algorithms

Monte Carlo approximations offer an alternative when enumeration is impossible. With this approach, obtaining a reasonable estimate of Monte Carlo error is of paramount importance. With this estimate, Monte Carlo approximations have the huge benefit over large sample approximations that the error may reduced (with a high probability) to a desired level of accuracy. In this setting, the independence

model is unusual in that (no waste) independent draws from the conditional distribution can easily be obtained Agresti et al. (1979); Patefield (1981). For this case, the Monte Carlo estimate of the conditional P-value is given by:

$$\tilde{P} = \frac{\sum_{i=1}^{N} I\{h_i \geq h_{obs}\}}{N},$$

where h_i is the value of h for the ith generated sample and N is the number of generated samples. The estimate of the Monte Carlo standard error is $\sqrt{\tilde{P}(1-\tilde{P})/N}$. As in the enumeration problem, obtaining samples from more complex models containing the independence model as a special case, by simulating under independence and rejecting the tables that do not meet extra requirements can be extremely inefficient. These methods are referred to as simulate and reject algorithms.

Broadly speaking, Monte Carlo algorithms for log-linear and logistic models fall into a mixture of a few categories: special cases where exact simulation is available (Agresti et al.; 1979; Patefield; 1981); simulate and reject algorithms (Mehta et al.; 2000); Markov chain algorithms that update a portion of the current state leaving the remainder fixed (Forster et al.; 1996; Smith et al.; 1996a,b; McDonald et al.; 1999; Diaconis and Sturmfels; 1998); or finally, correcting a close distribution using importance sampling (Booth and Butler; 1999). Unfortunately, with the exception of the special cases where exact sampling is available, no uniformly optimal approach exists. As discussed earlier, simulate and reject schemes can be extremely inefficient. Schemes that generate from the reference set with no waste, such as Casella and Wells , have a large computational overhead. Similarly, Markov chain Monte Carlo algorithms, such as Smith et al. (1996a) and versions of Diaconis and Sturmfels (1998) that perform (a version of) random scan Gibbs sampling also require substantial enumeration within each step. Markov chain approaches have the added problem that convergence control is a difficult

issue. Further, the Diaconis and Sturmfels (1998) algorithm, regardless of how it is implemented, often requires prohibitive computationally intensive preprocessing. For the remainder of this chapter we discuss key papers on Monte Carlo sampling conditional distributions for log-linear models. We focus on general purpose algorithms that apply to classes of log-linear models. A related area of literature we do not cover is decomposable log-linear models. Lauritzen (1996) gives a thorough account of these models as well as a Markov chain algorithm to fit them.

3.5 Algorithm of Casella and Wells

As discussed in the previous section, Casella and Wells presented a method for representing Γ as the zeros of a sequence of polynomials with sequential indeterminants (where, recall, an indeterminant is the argument of a function). Specifically, they represent the reference set as the solutions to[2]

$$\begin{aligned} p_1(y_1) &= 0 \\ p_2(y_1, y_2) &= 0 \\ &\vdots \\ p_n(y_1, \ldots, y_n) &= 0, \end{aligned}$$

for polynomials p_i. This suggests sampling y_1 from the set of solutions to $p_1(y_1) = 0$. Then, given y_1, sample y_2 from the zeros of $p_2(y_1, y_2) = 0$ and so on. An important attribute of the Groebner basis is that for any y_1 that satisfies $p(y_1) = 0$, there is at least one corresponding point, y_2, so that $p(y_1, y_2) = 0$, and at least one corresponding point, y_3, so that $p(y_1, y_2, y_3) = 0$, and so on. That is to say, generating from the reference set in this way produces no wasted samples. The

[2] In several cases, for this system of equations to form a Groebner basis, there must be more than n polynomials.

Table 3.2: A 3×3 table.

y_{11}	y_{12}	-	n_{1+}
y_{21}	y_{22}	-	n_{2+}
-	-	-	n_{3+}
n_{+1}	n_{+2}	n_{+3}	n

difficulty in this approach is choosing the correct sequential distributions to sample from.

We will generally have a range of possible values for each y_i under the conditional distribution, say $0 \leq y_i \leq n_i - 1$. For example in conditional logistic regression the n_i might be one plus the binomial sample sizes, or for contingency tables these might be calculated using margin constraints. With these bounds it will take at most n_1 operations to calculate the zeros of $p_1(y_1)$. Then, given y_1, it will take at most n_2 operations to calculate the zeros of $p_2(y_2)$ and so on. Therefore, once the Groebner basis has been constructed, it will take at most $n_1 + \ldots + n_n$ operations to generate an element from the reference set. In fact, it will always take fewer calculations for two reasons. First, as the zeros of $p(y_1)$ do not depend on any other y_i, they may be stored and thus one only needs to calculate them once. Secondly, some of the y_j will be will be completely determined by the previously generated variables.

The main drawback of this approach is the heavy computational burden required to calculate the Groebner basis. We now demonstrate that, in some cases, this basis (or at least a system of polynomials with sequential indeterminants whose zeros are the reference set) can be calculated by hand. Patefield (1981) describes an algorithm for sequentially sampling from the conditional distribution for the independence model for $I \times J$ tables.

We now describe Patefield's algorithm. Given the previously generated cells, cell y_{ij} is generated by the 2×2 table defined by the sums: $\sum_{k=i+1}^{I} y_{kj}$ (the cells below y_{ij}), $\sum_{l=j+1}^{J} y_{il}$ (the cells to the right of y_{ij}), $\sum_{k=i+1}^{I} \sum_{l=j}^{j} y_{kl}$ (the cells

below and to the right of y_{ij}), and $\sum_{k=i}^{I}\sum_{l=j+1}^{j} y_{kl}$ (the cells to the right and below y_{ij}). These sums can be calculated using the previously generated cells, $\{y_{kl}\}_{k=1,l=j-1}^{I} \cup \{y_{il}\}_{l=1}^{j-1}$ (the cells to the left or above y_{ij}) and the margins. We illustrate using the 3×3 table given in Table 3.2. Note that since the margins are fixed the upper left four entries determine the remaining entries. First, generate y_{11} from the hypergeometric distribution defined by the 2×2 table,

y_{11}	-	n_{1+}
-	-	$n_{2+} + n_{3+}$.
n_{+1}	$n_{+2} + n_{+3}$	

Next generate y_{12} from the hypergeometric distribution defined by the 2×2 table:

y_{12}	-	$n_{1+} - y_{11}$
-	-	$n - n_{1+} - n_{+1} + y_{11}$,
n_{+2}	n_{+3}	

then solve for $y_{13} = n_{1+} - y_{11} - y_{12}$. Then y_{21} is generated from the hypergeometric distribution defined by the 2×2 table:

y_{21}	-	n_{2+}
-	-	n_{3+} .
$n_{+1} - y_{11}$	$n_{+2} + n_{+3} - y_{12} - y_{13}$	

Then y_{22} is generated from the hypergeometric distribution defined by the 2×2 table:

y_{22}	-	$n_{2+} - y_{21}$
-	-	$n - n_{1+} - n_{2+} - n_{+1} + y_{11} + y_{12}$,
$n_{+2} - y_{12}$	$n_{+3} - y_{13}$	

and solve for the remaining positions. Note that, following the sequential generation,

$$\begin{aligned}
\max(n_{+1} - n_{1+}, 0) \leq\ & y_{11} && \leq \min(n_{+1}, n_{1+}) \\
\max(n_{+2} - n + n_{+1} + n_{1+} - y_{11}, 0) \leq\ & y_{12} && \leq \min(n_{1+} - y_{11}, n_{+2}) \\
& y_{13} && = n_{1+} - y_{11} - y_{12} \\
\max(n_{+1} - y_{11} - n_{3+}, 0) \leq\ & y_{21} && \leq \min(n_{+1} - y_{11}, n_{2+}) \\
\max(n_{+2} - y_{11} - n + n_{1+} + n_{2+} + n_{+1} - y_{11} - y_{12}, 0) \leq\ & y_{22} && \leq \min(n_{+2} - y_{12}, n_{2+} - y_{21}) \\
& y_{23} && = n_{2+} - y_{21} - y_{22} \\
& y_{31} && = n_{+1} - y_{11} - y_{21} \\
& y_{32} && = n_{+2} - y_{12} - y_{22} \\
& y_{33} && = n_{+3} - y_{13} - y_{23}
\end{aligned}$$

These equations can be represented by a system of polynomial equations. To illustrate how this is done for the inequalities, note, for example, that

$$\max(n_{+1} - y_{11} - n_{3+}, 0) \leq y_{21} \leq \min(n_{+1} - y_{11}, n_{2+})$$

is equivalent to the four inequalities

$$\begin{aligned}
y_{21} &\geq 0 \\
y_{21} &\geq n_{+1} - y_{11} - n_{3+} \\
y_{21} &\leq n_{+1} - y_{11} \\
y_{21} &\leq n_{2+}
\end{aligned}$$

Therefore these inequalities are equivalent to the simultaneous zeros of

$$p_1 = \prod_{i=n_{+1}-n_{3+}}^{n_{+1}} (y_{11} + y_{21} - i) \text{ and } p_2 = \prod_{i=0}^{n_{2+}} (y_{21} - i).$$

Further note that, the simultaneous zeros of p_1 and p_2 are also the zeros of $p_1^2 + p_2^2$, provided the polynomials are on the fields of real or rational numbers. Therefore, using this technique on the other inequalities, we can calculate the Groebner basis for this problem without any computational tools. We note, however, the Groebner basis is not of as much practical interest as the sequential generating algorithm we used to create it.

The independence model is not the only model where closed form expressions exist for the Groebner basis. Consider the model of complete symmetry for $I \times I$ contingency tables (Agresti; 1990). The complete symmetry multinomial model simply specifies that the probability of an i, j response is the same as the probability of a j, i response. Here the sufficient statistics are the sum of the symmetrically opposite cells. That is, if y_{ij} represent the cells of the table, then the sufficient statistics are $s_{ij} = y_{ij} + y_{ji}$. As $s_{ji} = s_{ij}$ we adopt the convention of always placing the smaller subscript first (i.e. s_{12} instead of s_{21}). Note that since $s_{ii} = y_{ii}$, the conditional distribution for the symmetry model fixes the diagonal. The relations

$$\begin{aligned} 0 \leq y_{ij} \leq s_{ij} \quad & \text{for } i < j, \\ y_{ij} = s_{ij} \quad & \text{for } i = j, \\ y_{ij} = s_{ji} - y_{ji} \quad & \text{for } i > j, \end{aligned}$$

completely define the reference set. This is equivalent to the solution set of the system of polynomial equations

$$\begin{aligned}
y_{11} - s_{11} &= 0 \\
\prod_{i=0}^{s_{12}} (y_{12} - i) &= 0 \\
&\vdots \\
\prod_{i=0}^{s_{1I}} (y_{1I} - i) &= 0 \\
y_{21} + y_{12} - s_{12} &= 0 \\
y_{22} - s_{22} &= 0 \\
\prod_{i=0}^{s_{23}} (y_{23} - i) &= 0 \\
&\vdots \\
y_{II} - s_{II} &= 0.
\end{aligned}$$

A third model for $I \times I$ tables, quasi-symmetry, has the sum of symmetrically opposite cells and the margins, as its sufficient statistics. Thus, the quasi-symmetry model contains independence and complete symmetry as special cases. If we let $p_{11}^{ind}(y_{11}), \ldots, p_{II}^{ind}(y_{11}, \ldots, y_{II})$ and $p_{11}^{sym}(y_{11}), \ldots, p_{II}^{sym}(y_{11}, \ldots, y_{II})$ be Groebner bases for the independence and symmetry models respectively, then the simultaneous zeros of

$$\begin{aligned}
p_{11} &= p_{11}^{ind}(y_{11})^2 + p_{11}^{sym}(y_{11})^2 \\
p_{12} &= p_{12}^{ind}(y_{11}, y_{12})^2 + p_{12}^{sym}(y_{11}, y_{12})^2 \\
&\vdots \\
p_{II} &= p_{II}^{ind}(y_{11}, \ldots, y_{II})^2 + p_{II}^{sym}(y_{11}, \ldots, y_{II})^2
\end{aligned}$$

Table 3.3: Cross-classification of husband and wife's ratings of sexual fun.

	Wife's Rating			
Husband	Never or occasionally	Fairly often	Very often	Almost always
N	7	7	2	3
F	2	8	3	7
V	1	5	4	9
A	2	8	9	14

Source: (Agresti; 1990)

define the reference set for the quasi-symmetry model. A flaw in this approach is that the p_i do not form a Groebner basis. Therefore, for example, a y_{11} may exists such that $p_{11}(y_{11}) = 0$; however, given this y_{11}, no y_{12} exists so that $p_{12}(y_{11}, y_{12}) = 0$. For some tables, the number of these occurrences is small enough so that generating from the zeros of these polynomials is useful. For example, consider the data given in Table 3.3 (Agresti; 1990) which cross-classifies the paired ratings of sexual fun for (heterosexual) married couples. Generating from the zeros of the p_{ij}, with probabilities proportional to

$$\exp(\hat{\mu}_{ij} \log y_{ij} - \hat{\mu}_{ij})/y_{ij}!,$$

where $\hat{\mu}_{ij}$ is the fitted value for cell i, j, resulted in a 53% acceptance rate. This rate can be much lower for large dimensional tables. For example, the acceptance rate was 0% out of 10,000 simulations for the sparse 8×8 table presented in Smith et al. (1996b). In general, the process of avoiding calculating the Groebner basis for a particular model by combining the Groebner bases for sub-models will not produce a system of polynomials rich enough for effective enumeration or simulation.

3.6 Random Scan Gibbs Sampling

Forster et al. (1996); Smith et al. (1996a,b) and McDonald et al. (1999) show how to use random scan Gibbs sampling to draw Markovian samples from $f(\mathbf{y}|\mathbf{s})$.

Their approach is as follows. Assume $\begin{bmatrix} \mathbf{X}' & \mathbf{Z}' \end{bmatrix}'$ is an $n \times n$ invertible matrix (that is, the alternative model is saturated). Define

$$\begin{bmatrix} \mathbf{X}' & \mathbf{Z}' \end{bmatrix} \mathbf{y} = \begin{bmatrix} \mathbf{s} & \mathbf{z} \end{bmatrix}.$$

Sampling from $f(\mathbf{z}|\mathbf{s})$ is equivalent to sampling from $f(\mathbf{y}|\mathbf{s})$. It is preferable to work with the $\mathbf{z}$ rather than the $\mathbf{y}$ as the requirement $\mathbf{s} = \mathbf{s}_{obs}$ reduces the dimension of $\mathbf{y}$. For example, in a independence model for a 5×5 table, the sufficient statistics are the margins and hence 16 (the rank of $\mathbf{X}$) cells determine the remaining 9. One might be inclined to say the dimension of the free cells is always the rank of $\mathbf{X}$. However, this is not always the case because the cells have to be non-negative integers. Thus in general, one can only say that there are no more than rank($\mathbf{X}$) free cells. This is most evident in exact logistic regression where often only the observed data satisfies the conditions of the reference set, regardless of the rank of $\mathbf{X}$.

Forster et al. (1996) show how to calculate the exact distribution of the full conditionals of $f(\mathbf{z}|\mathbf{s})$ and implement random scan Gibbs sampling (see Section 2.5). Calculating the full conditional corresponds to a smaller enumeration problem.

In models with significant structure, this algorithm suggests another algorithm, which they implement in square contingency tables in Smith et al. (1996a). To illustrate, consider the independence model. Their algorithm randomly selects a tetrad[3] , say $y_{ij}, y_{i'j}, y_{ij'}, y_{i'j'}$, and updates those cells with probabilities proportional to $(y_{ij}!y_{i'j}!y_{ij'}!y_{i'j'}!)^{-1}$; that is according to the hypergeometric distribution defined by the margins.

[3] A tetrad is the four cells defined where two different rows and two separate columns intersect.

For the complete symmetry model, this algorithm randomly chooses a pair of symmetrically opposite cells, say y_{ij} and y_{ji} and updates them with probability $(y_{ij}!y_{ji}!)^{-1}$ where their sum is the same as the observed data. Smith et al. (1996a) discuss several examples of square contingency tables including quasi-independence, quasi-symmetry and a model that contains the uniform association and quasi-symmetry as special cases. Although the method appears to be quite efficient, their papers include no proof of irreducibility of the resulting Markov chains. In fact, Mehta et al. (2000) show that for an exact logistic regression example, the algorithm of Smith et al. (1996a) can produce reducible chains. In the next section, we show how the work of Diaconis and Sturmfels (1998) may be used to prove irreducibility of some of Smith et al.'s algorithms.

3.7 Random Walk Metropolis Hastings

A seminal paper by Diaconis and Sturmfels (1998) drew the first connection between exact conditional analysis and computational algebraic geometry. Their algorithm is a random walk Metropolis algorithm on the reference set. In many cases, their algorithm results in a minor perturbation of the current table, and thus is often said to produce "local moves" within the reference set. For specific models this approach had been suggested prior to Diaconis and Sturmfels (1998). These are reviewed in Fienberg et al. (1999), and include unpublished manuscripts by Darroch and Glonek and well known papers by Besag and Clifford (1989) and Guo and Thompson (1992). A review paper by Bunea and Besag (2000) also uses Markov chains with local moves.

Consider $\mathbf{X}'$ as a linear transformation from $\mathbb{Z}^n$ to $\mathbb{Z}^p$. This will be the case provided $\mathbf{X}'$ has integer valued entries. Let $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k$ be a basis for the null space of $\mathbf{X}'$. Then the $\mathbf{f}_i$ satisfy, $\mathbf{X}'\mathbf{f}_i = 0$ and any for any two elements $\mathbf{y}_1$ and $\mathbf{y}_2$ in Γ, $\mathbf{y}_2 = \sum_{i=1}^{k} a_i\mathbf{f}_i + \mathbf{y}_1$ for some integers a_i. Thus it is possible to move between any two vectors in the reference set by adding or subtracting integer multiples of

the basis elements. This suggest two methods. The first is to simulate the a_i from some distribution on the k dimensional integers and then take $\sum_{i=1}^{k} a_i \mathbf{f}_i + \mathbf{y}_{obs}$ as the simulated data point for a Metropolis step (or importance sampling step). Notice that the simulated data will have $\mathbf{s}$ as its sufficient statistic (as $\mathbf{X}'\mathbf{y}_{obs} = \mathbf{s}_{obs}$) and every element of the reference set can be generated this way. However the simulated data may have negative entries. Diaconis and Sturmfels (1998) claim to have tried this method, but found that choosing a distribution for the a_i that produced a reasonable number of simulated vectors with non-negative entries was very difficult.

The second idea is to use a random walk Metropolis algorithm (see Section 2.5). That is, if $\mathbf{y}_1$ is the current state of the chain, then take $\mathbf{y}_2$ to be $\mathbf{y}_1 + a\mathbf{f}_i$ where i is chosen randomly and $a = \pm 1$ each with probability one half. The problem with this algorithm is that for a given $\mathbf{y}_1$ and $\mathbf{y}_2$, there may be no way to get from one to the other by adding (or subtracting) multiples of the $\mathbf{f}_i$ without leaving the reference set. That is, the chain is not connected. Diaconis and Sturmfels (1998) address this issue by adding on to the basis. They define a *Markov basis*, in this context, to be a collection $\{\mathbf{f}_i\}_{i=1}^{k}$ such that for any $\mathbf{y}_1$, $\mathbf{y}_2$ in Γ, it is possible to get from $\mathbf{y}_1$ to $\mathbf{y}_2$ (and vice versa) by adding unit multiples of the $\mathbf{f}_i$ in such a way that the partial sums never leave Γ. As a result, the Markov basis approach guarantees an irreducible Markov chain. Diaconis and Sturmfels (1998) show how to create a Markov basis using the tools of computational algebraic geometry.

In models with simple structure, the Markov basis takes on a simple form. For example, for the independence model for an $I \times J$ table, the Markov basis contains the $\binom{I}{2}$ $\binom{J}{2}$ tetrads with entries $\begin{matrix} +1 & -1 \\ -1 & +1 \end{matrix}$ and zeros in the other cells. For the

symmetry model the Markov basis elements look like $\begin{matrix} 0 & +1 \\ -1 & 0 \end{matrix}$ for symmetrically opposite cells and zeros elsewhere.

This algorithm is similar to the algorithm presented in Smith et al. (1996a) which we refer to as "the SFM algorithm". We draw the connection with SFM's algorithm through the independence and symmetry examples. Notice that, for both of these models, the Markov basis moves are contained in the collection of possible moves in one iteration of the SFM algorithm. Therefore, at least for these models, irreducibility of SFM's is guaranteed. In fact, in these cases, SFM's random scan Gibbs sampler is exactly the same as an algorithm that randomly selects a Markov basis member, say $\mathbf{f}_i$, then determines for which integer values a, $\mathbf{y}_1 + a\mathbf{f}_i$ is in Γ and selects one of these elements with probabilities proportional to $(\mathbf{y}_1 + a\mathbf{f}_i)!^{-1}$. This method was suggested in Diaconis and Sturmfels (1998). However, the connection with SFM's algorithm was not made. In the situations where SFM's algorithm does not produce an irreducible chain, their collection of moves does not contain all of the Markov basis elements.

As with Casella and Wells, the algebraic computations required to get the Markov basis are substantial, especially in problems with ordinal scores or covariates, such as linear by linear association models and exact logistic regression. We have found cases where the number of elements of the Markov basis actually exceeds the number of elements of the reference set! It is yet to be shown that the Markov basis approach can solve problems that cannot be solved otherwise.

3.8 Discussion

Although this chapter focused primarily on categorical data analysis, conditional analysis also applies in continuous settings. In these cases, excessive conservatism is not a problem. In fact, as argued in the introduction, most of

the standard normal theory tests can be viewed as conditional tests in which the conditioning is irrelevant.

Some applications of conditional analysis for continuous problems are given in Butler et al. (1999) who tackle the computational issues using a saddle-point/importance sampling approach. In Chapter 7 we discuss the possibility of extending the Markov chain algorithm given in the next chapter to this setting. Alternatively, in some cases, methods similar to Casella and Wells might be employed. Here the support of the conditional distribution is often a compact subset of $\mathbb{R}^k$ for some k. If this subset could be demonstrated as the zeros of a collection of polynomials (such subsets are called varieties), a Groebner basis could be constructed to describe this variety. The basis could be used to uniformly sample from the variety which could then be used as an importance distribution.

CHAPTER 4
AN MCMC ALGORITHM FOR APPROXIMATING EXACT CONDITIONAL PROBABILITIES

4.1 Introduction

In this chapter we present new methodology for performing the exact conditional tests described in Chapter 3. We maintain the notation of that chapter throughout. As before, our key example will be the data given in Table 1.1 which represents the cross-classified ratings of 118 tumors by two pathologists. A previously discussed model of interest assumes the cell counts, y_{ij}, are independent Poisson random variables with means, μ_{ij}, satisfying independence: $\log(\mu_{ij}) = \beta + \beta_i^1 + \beta_j^2$. A perhaps more reasonable model, that accounts for the ordinal nature of the classifications is the uniform association model, $\log(\mu_{ij}) = \beta + \beta_i^1 + \beta_j^2 + \beta^3 ij$. If either of these models represents the null model for a lack of fit test, then all probability calculations under the null hypothesis depend on the values of $\beta, \beta_i^1, \beta_j^2$ and β^3 (in the uniform association model). Conditional inference eliminates these parameters by conditioning on their sufficient statistics. The benefit of conditional inference lies in the fact that the resulting tests are exact (see the arguments surrounding equation (3.3)). That is, the true type one error rate (obtained from the unconditional distribution) is no larger than the nominal for all values of the nuisance parameters.

With Poisson sampling the conditional distribution is proportional to $1/\prod_{ij} y_{ij}!$, as derived in (3.8). As discussed in the previous chapter neither model leads to efficient enumeration. The reference set for the independence model contains too many elements while the reference set for the uniform association

model is too complicated. Further we note that this problem is not one that will disappear with increases in processor speed. Gail and Mantel (1977), Good (1976) and Good (1979) give approximations for the number of operations necessary to enumerate the reference set for the independence model. In particular, it is shown that the number of operations grows faster than any polynomial in the margin counts. Thus even if enumerating the 12 billion tables is tractable, one need only mildly increase the margin counts to come up with a reference set that cannot be enumerated.

Analytical methods for approximating conditional distributions arising in contingency table analysis have been studied extensively. These methods typically involve very little computation and so offer a useful alternative in situations where complete enumeration of the reference set is not feasible. The most widely used is the chi-squared limiting distribution for the Pearson chi-squared and likelihood ratio statistics. However, for sparse data, P-value estimates obtained using the chi-squared distribution can differ greatly from exact conditional P-values. For example, the chi-squared P-value for testing lack-of-fit of the uniform association model to the data in Table 1.1, using the likelihood ratio statistic, is .37. This value is drastically different from the exact conditional P-value, .044. More accurate normal approximations based on expansions for the conditional moments of the likelihood ratio statistic have been studied by McCullagh (1986) and Paul and Deng (2000). Another approach which has received a considerable amount of attention is saddlepoint approximation. For example, Davison (1988) applies Skovgaard's (1987) saddlepoint approximation and Pierce and Peters (1992) apply Barndorff-Nielsen's (1986) r^* formula to approximate conditional distributions in discrete exponential family models. More recently, Strawderman and Wells (1998) use the saddlepoint formula of Lugannani and Rice (1980) to approximate the exact conditional distribution for testing conditional independence

versus a common odds-ratio in a $2 \times 2 \times K$ table. These methods are typically extremely accurate in the problems in which they can be applied. However, so far, saddlepoint methods have been limited to tests concerning a scalar parameter. So, for example, no saddlepoint approximation is yet available for testing lack-of-fit of the uniform association model to the data in Table 1.1. A quite different analytical approach to conditional inference developed by Waterman and Lindsay (1996), based on projected scores, also produces extremely accurate answers in certain problems.

This chapter introduces a new Markov chain Monte Carlo algorithm for approximating conditional P-values for log-linear models based on the importance sampling approach of Booth and Butler (1999). In particular we use Algorithm 2.2 to perform local moves within the reference set, such as: Forster et al. (1996), Smith et al. (1996a), Smith et al. (1996b), McDonald et al. (1999), and Diaconis and Sturmfels (1998), while maintaining the flexibility of Booth and Butler's algorithm.

4.2 Monte Carlo Approximations

In this section, we outline Booth and Butler's algorithm. They approximate the conditional P-value using importance sampling with rounded normal deviates. Booth and Butler use an importance sampling algorithm (see Section 2.3) for draws from a crude normal approximation to $f(\mathbf{y}|\mathbf{s}_{obs})$.

Booth and Butler construct their importance distribution as follows. As $\mathbf{Y}$ is Poisson with mean vector $\boldsymbol{\mu}$, $\mathbf{Y}$ should be approximately normal with mean vector $\boldsymbol{\mu}$ and variance covariance matrix $D(\boldsymbol{\mu})$, where $D(\boldsymbol{\mu})$ is a diagonal matrix with diagonal $\boldsymbol{\mu}$. Recall that $\mathbf{X}$ is an $n \times p$ matrix with rank $p \leq n$. Hence we can partition $\mathbf{X}^t = \begin{bmatrix} \mathbf{X}_1^t & \mathbf{X}_2^t \end{bmatrix}$ so that $\mathbf{X}_2^t$ $(p \times p)$ is full rank. This may require some rearrangement of the rows of $\mathbf{X}$. Let $\mathbf{Y}^t = (\mathbf{Y}_1^t \ \mathbf{Y}_2^t)$ and $\boldsymbol{\mu}^t = (\boldsymbol{\mu}_1^t \ \boldsymbol{\mu}_2^t)$ be the

corresponding partitions of $\mathbf{Y}^t$ and $\boldsymbol{\mu}^t$. Then since

$$\mathbf{S} = \mathbf{X}^t\mathbf{Y} = \mathbf{X}_1^t\mathbf{Y}_1 + \mathbf{X}_2^t\mathbf{Y}_2$$

and hence $\mathbf{Y}_2 = (\mathbf{X}_2^t)^{-1}(\mathbf{S} - \mathbf{X}_1^t\mathbf{Y}_1)$. It follows that $\mathbf{Y}_1$ and $\mathbf{S}$ determine $\mathbf{Y}_2$.

Then note,

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{S} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{X}_1^t & \mathbf{X}_2^t \end{bmatrix} \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \end{bmatrix},$$

is approximately normal with mean

$$\begin{bmatrix} \boldsymbol{\mu}_1 \\ \mathbf{X}^t\boldsymbol{\mu} \end{bmatrix}$$

and variance

$$\begin{bmatrix} D(\boldsymbol{\mu}_1) & D(\boldsymbol{\mu}_1)\mathbf{X}_1 \\ \mathbf{X}_1^t D(\boldsymbol{\mu}_1) & \mathbf{X}_2^t D(\boldsymbol{\mu}_2)\mathbf{X}_2 \end{bmatrix}.$$

Then using the conditional moments for the multivariate normal (for example in Searle; 1997, Section 4 of Chapter 2) we obtain $\mathbf{Y}_1|\mathbf{S} = \mathbf{s}$ is approximately normal with mean

$$\boldsymbol{\mu}_1 + \mathbf{X}_1^t D(\boldsymbol{\mu}_1)(\mathbf{X}_2^t D(\boldsymbol{\mu}_2)\mathbf{X}_2)^{-1}(\mathbf{s} - \mathbf{X}^t\boldsymbol{\mu}) \tag{4.1}$$

and variance

$$\boldsymbol{\Sigma} = D(\boldsymbol{\mu}_1) + D(\boldsymbol{\mu}_1)\mathbf{X}_1(\mathbf{X}_2^t D(\boldsymbol{\mu}_2)\mathbf{X}_2)^{-1}\mathbf{X}_1^t D(\boldsymbol{\mu}_1). \tag{4.2}$$

Notice that the conditional variance does not depend on $\mathbf{s}$. Booth and Butler use this normal distribution to simulate integer vectors approximately possessing mean (4.1) and variance (4.2). We describe how this is done in the next section. In particular we shall see that the order in which the elements of $\mathbf{Y}_1$ are listed (we call this the update order) changes the normal approximation. Notice that some value of $\boldsymbol{\mu}$ is needed for simulation, say $\hat{\boldsymbol{\mu}}$. The fitted values, $\hat{\boldsymbol{\mu}}$ say, should be a

reasonable approximation to $E[\mathbf{Y}_1|\mathbf{S} = \mathbf{s}_{obs}]$. Further, note that the fitted values satisfy $\mathbf{X}^t\hat{\boldsymbol{\mu}} = \mathbf{s}_{obs}$ hence simplify (4.1) considerably.

The ability of the normal importance distribution to approximate the covariance structure for the conditional distribution makes their algorithm very efficient in a wide range of problems. A drawback to this approach lies in the support of the importance distribution properly containing the reference set, requiring the algorithm to throw away all generated samples not in the reference set. We have found instances where in excess of 99% of the generated samples must be thrown out. The main result of this Chapter is a Metropolis Hastings algorithm (of the type listed in Algorithm 2.2) that leaves a portion of the current table fixed at each iteration and applies Booth and Butler's normal approximation to the remainder while randomizing the update order. This algorithm can greatly increase the number of tables generated with the correct sufficient statistics and hence extend the applicability of the normal approximation. It can be seen as a hybrid approach between Booth and Butler's algorithm and the local update algorithms (Forster et al.; 1996; Smith et al.; 1996a,b; McDonald et al.; 1999; Diaconis and Sturmfels; 1998).

4.3 Choosing the Candidate Transition Kernel

We now show how we adapt Booth and Butler's normal candidate to produce a candidate transition kernel on the reference set which we use for a Metropolis Hastings algorithm. Let $\mathbf{P}$ be an $(n-p) \times (n-p)$ permutation matrix and let $\mathbf{P}^t = \begin{bmatrix} \mathbf{P}_1^t & \mathbf{P}_2^t \end{bmatrix}$ be a partition of $\mathbf{P}$ with $\mathbf{P}_1^t$ $(n-p) \times k$ for some $1 \leq k \leq n-p$. We use $\mathbf{P}$ to randomize the update order and the partition determines which cells are to be updated. More specifically, we generate k and $\mathbf{P}$ randomly and update the cells corresponding to $\mathbf{P}_1\mathbf{Y}_1$ and leave the cells corresponding to $\mathbf{P}_2\mathbf{Y}_2$ fixed. Notice that $\mathbf{P}\mathbf{Y}_1$ will be approximately normal with mean $\mathbf{P}\boldsymbol{\mu}_1$ (assuming we use $\hat{\boldsymbol{\mu}}$

for simulation) and variance

$$\mathbf{P}\boldsymbol{\Sigma}\mathbf{P}^t = \begin{bmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{12}^t & \mathbf{V}_{22} \end{bmatrix} \tag{4.3}$$

where $\mathbf{V}_{11}$ is a $k \times k$ matrix. Let $\mathbf{Y}_1^\star = \mathbf{P}_1\mathbf{Y}_1$ and $\mathbf{Y}_2^\star = \mathbf{P}_2\mathbf{Y}_1$. Then $\mathbf{Y}_1^\star|\mathbf{Y}_2^\star = \mathbf{y}_2^\star, \mathbf{S} = \mathbf{s}$ is approximately normal with mean

$$\mathbf{P}_1\boldsymbol{\mu}_1 + \mathbf{V}_{12}\mathbf{V}_{22}^{-1}(\mathbf{P}_2\boldsymbol{\mu}_1 - \mathbf{Y}_2^\star) \tag{4.4}$$

and variance

$$\mathbf{V}_{11} + \mathbf{V}_{12}^t\mathbf{V}_{22}^{-1}\mathbf{V}_{12}. \tag{4.5}$$

A random vector from the approximate normal distribution of $\mathbf{Y}_1^\star$ given $(\mathbf{s}, \mathbf{y}_2^\star)$ can be generated by sequentially simulating univariate normals as follows:

$$\begin{aligned} Y_{11}^\star &\sim N(m_1(\boldsymbol{\mu}), v_1(\boldsymbol{\mu})) \\ Y_{1j}^\star &\sim N(m_j(\boldsymbol{\mu}, y_{11}^\star, y_{12}^\star, \ldots, y_{1(j-1)}^\star), v_j(\boldsymbol{\mu}))) \quad j = 2, \ldots, k. \end{aligned}$$

where the $y_{1j}^\star$ are the elements of $\mathbf{y}_1^\star$. Formulas for the sequential means and variances (m_i and v_i) are given in Searle (1997). In fact, these sequential calculations can be carried out using only univariate calculations. In particular no matrix inversion is required (see Booth and Butler; 1999, appendix).

This procedure requires some modification because the components of $\mathbf{y}$ must be integers. This can be accomplished by rounding each normal variate after it is generated and conditioning on this rounded value in the subsequent calculations. As we will see this sequential rounding greatly simplifies specification of the candidate transition kernel; as opposed to a similar procedure that generates a multivariate normal vector and then rounds it.

Assume the chain is at the current state (table) $\mathbf{z}^t = \begin{bmatrix} \mathbf{z}_1^t & \mathbf{z}_2^t \end{bmatrix} \in \Gamma$. Then $\mathbf{y}_1^\star$ is generated from the rounded normal approximation with $\mathbf{y}_2^\star = \mathbf{P}_2\mathbf{z}_1$. The

candidate table is $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1^t & \mathbf{y}_2^t \end{bmatrix}$ where

$$\mathbf{y}_1 = \mathbf{P}^t \begin{bmatrix} \mathbf{y}_1^\star \\ \mathbf{y}_2^\star \end{bmatrix},$$

(where we recall that the transpose of a permutation matrix is its inverse) and

$$\mathbf{y}_2 = (\mathbf{X}_2^t)^{-1}(\mathbf{s} - \mathbf{X}_1^t \mathbf{y}_1).$$

Notice that $\mathbf{y}$ will satisfy: $\mathbf{X}^t\mathbf{y} = \mathbf{s}$, $\mathbf{P}_2\mathbf{y}_1 = \mathbf{P}_2\mathbf{z}_1$. That is, the cells corresponding to $\mathbf{P}_2\mathbf{z}_1$ stay the same. The probability of generating $\mathbf{y}$ when the current state is $\mathbf{z}$ is given by

$$g(\mathbf{y}|\mathbf{z}, \mathbf{P}, k) \propto \prod_{i=1}^{k} \left[\Phi\left(\frac{y_{1i}^\star + .5 - m_i}{v_i} \right) - \Phi\left(\frac{y_{1i}^\star - .5 - m_i}{v_i} \right) \right]. \tag{4.6}$$

To account for possible negative entries, or if any of the entries of $\mathbf{y}_2$ are not integers we set $g(\mathbf{y}|\mathbf{z}, \mathbf{P}, k) = 0$ if $\mathbf{y} \notin \Gamma$.

We note here that a major strength of the algorithm proposed in this chapter is its simplicity. More accurate approximations for the conditional covariance structure have been developed, for example, by Waterman and Lindsay (1996). However, their approximation is considerably more complicated than the one we use and we seriously doubt that an improvement in the covariance approximation would result in a significant change in the performance of our algorithm. More generally, one might consider better approximations for the entire conditional distribution, such as the saddlepoint density Daniels (1954). However, it is not at all clear how one can simulate directly from a multivariate saddlepoint density and hence that approximation is not useful in this context.

4.4 The Algorithm

We now outline the full algorithm. First notice we are running the version of the Metropolis Hastings algorithm given in Algorithm 2.2 where we average our chain over $\mathbf{P}$ and k. Let $\mathbf{z}$ be the current value of the chain. For the first iteration it can simply be the data values themselves. Recall $\mathbf{s}_{obs}$ is the observed sufficient statistic. Let the variables *iteration* and *successes* be counters initially started at 0.

Algorithm 4.1 An MCMC Algorithm for Log-linear Models

1. Generate a permutation matrix $\mathbf{P}$.
2. Generate the number of table entries to be updated, k.
3. Partition $\mathbf{P} = \begin{pmatrix} \mathbf{P}_1 \\ \mathbf{P}_2 \end{pmatrix}$ where $\mathbf{P}_1$ has k rows.
4. Generate $\mathbf{y}_1^* = \mathbf{P}_1\mathbf{y}_1$ using the rounded normal candidate with mean (4.4) and variance (4.5). Leave $\mathbf{y}_2^* = \mathbf{P}_2\mathbf{z}_1$ fixed.
5. If any of the components of $\mathbf{y}_1^*$ are negative then go to step 11 with the chain staying at $\mathbf{z}$. Otherwise go to step 6.
6. Calculate $\mathbf{y}_1 = \mathbf{P}^t \begin{pmatrix} \mathbf{y}_1^* \\ \mathbf{y}_2^* \end{pmatrix}$ and $\mathbf{y}_2 = (\mathbf{X}_2^t)^{-1}(\mathbf{s} - \mathbf{X}_1^t\mathbf{y}_1)$.
7. If any of the components of $\mathbf{y}_2$ are negative then go to step 11 with the chain staying at $\mathbf{z}$. Otherwise go to step 8.
8. Calculate the Metropolis Hastings transition probability $\rho(\mathbf{y}|\mathbf{z}, \mathbf{P}, k)$. The candidate transition kernel $g(\mathbf{y}|\mathbf{z}, \mathbf{P}, k)$ is defined in (4.6) The stationary mass function is $f \propto (\prod y_i!)^{-1}$.
9. Generate a random uniform variate u.
10. If $u < \rho(\mathbf{y}|\mathbf{z}, \mathbf{P}, k)$ the chain moves to state $\mathbf{y}$. Otherwise the chain remains at $\mathbf{z}$.
11. Increment *iterations* by 1.
12. Calculate the lack-of-fit statistic h of the current value of the chain.
13. If $h \geq h_{obs}$ increment *successes* by 1.

A couple of notes are in order. First, the updated p-value estimate is given by *successes/iterations*. Later we discuss quantifying the Monte Carlo error in this approximation. Secondly, step 8 requires calculating $g(\mathbf{z}|\mathbf{y}, \mathbf{P}, k)$, the probability of going in reverse from $\mathbf{y}$ to $\mathbf{z}$ given $\mathbf{P}$ and k. This is accomplished by simply calculating the sequential means as if $\mathbf{z}$ was generated and $\mathbf{y}$ was the current state.

The sequential variances stay the same as the ones for going from $\mathbf{z}$ to $\mathbf{y}$. These means, variances and $\mathbf{z}_1^* = \mathbf{P}_1\mathbf{z}_1$ are then plugged into (4.6) to obtain $g(\mathbf{z}|\mathbf{y}, \mathbf{P}, k)$.

There are several ways to generate k and $\mathbf{P}$ in steps 1 and 2. For the examples in this chapter, $\mathbf{P}$ was generated by randomly sorting the rows of an identity matrix. However, $\mathbf{P}$ could also be chosen systematically or otherwise. Similarly, k was generated from a binomial distribution, though other distributions or systematic schemes would also work. The key requirement is that the resulting chain is irreducible. This is the case if k is generated as a binomial with total number of trials equal to $n - p$, the number of rows of $\mathbf{X}_1$, since it is then possible to move from any table in the reference set to any other in one iteration. As expected the dependence in the chain is typically less severe the larger k is on average. However, the acceptance rate in the Metropolis step is lower for larger k. The best choice of k is as large as possible while still generating a reasonable number of tables in the reference set. If k is generated binomially, adaptively increasing or decreasing the success probability allows one to achieve this balance.

There are several tricks one can use to speed up the algorithm. For example, $(\mathbf{X}_2)^{-1}$ and $\mathbf{\Sigma}$ do not change from one iteration to the next and need only be calculated once. Secondly, in large dimensional tables (where this algorithm is useful), it is necessary to use a low binomial success probability so that few cells are updated on average per iteration. As the conditional variances (4.5) do not depend on the current state, the ones that will occur most frequently can be calculated prior to running the algorithm. Another possible solution, if computing time is a problem, is to choose $\mathbf{P}$ and its partition out of a smaller set than all possible combinations and preprocess the conditional variances for all of the elements of that set. Provided the set included a full update, irreducibility will not be affected.

As a last note, the algorithm generally seems to work better if the random variables generated in step 4 are generated from a t distribution rather than a normal. That is, the sequential means and variances, m_i and v_i, are created in the same manner except that each y_{ij}^* is generated as $m_i + t\sqrt{v_i}$ where t is a random draw from the t distribution. This requires modifying (4.6) accordingly. The inadequacy of the normal candidate transition kernel lies in the candidate transition probabilities of certain tables being exceptionally close to 0 (or rounded to 0 by limits of computer precision) in extremely sparse tables. For all practical purposes such chains are reducible; thus yielding a biased P-value estimate! The heavier tailed t distribution increases the candidate transition probabilities of such tables thus eliminating the bias at the expense of slightly slower convergence. The choice of degrees of freedom is arbitrary, however, one may feel more secure with a P-value estimate based on lower degrees of freedom. For all of the examples in this paper, 3 degrees of freedom were used.

Interestingly, some have argued the issue of burn-in is not technically a problem for Markov chain Monte Carlo algorithms for conditional tests. Besag and Clifford (1989) note that, under the null hypothesis, the observed data is an exact draw from the stationary distribution. This being the case, if the chain is started at the observed data, under the null hypothesis, the chain is at stationarity. This argument is unsatisfactory as it does not separate the random mechanism that generated the observed data from the MCMC simulations used to fit the data.

4.5 Examples

To test the algorithm it was applied to several different examples. In each example we assessed lack-of-fit of the null model using the deviance, that is, the likelihood ratio test versus a saturated model. Other measures of lack-of-fit could have been used just as easily but are not considered here because the key issue is the ability to efficiently sample from the null distribution. It should be noted,

however, that a drawback of likelihood ratio tests for non-saturated alternatives is the need to calculate the fitted values under the alternative. This can be circumvented by using a test statistic that does not require the fitted values under the alternative model, such as the efficient score statistic (McCullagh and Nelder; 1989, page 393).

The algorithm was tested on models of independence, quasi independence, quasi symmetry, and uniform association, as well as a Poisson log-linear model, and a capture/recapture model. The data and models for the Poisson log-linear and the capture/recapture examples can be found in Tables 4.4 and 4.5. The specification of the remaining the models can be found in Agresti (1990). Table 4.1 compares the results of the χ^2 approximation, the Monte Carlo estimate of the exact conditional P-value from the Metropolis algorithm and the exact conditional P-value. When the exact conditional P-value is unknown a Monte Carlo estimate based on another algorithm is given, except in Examples 5 and 6 where other reliable estimates were not available for comparison. The time until convergence was minimal for examples 1, 2, and 3 (a few minutes or less). Example 4 took around 30 minutes while examples 5 and 6 both took several hours to complete. In the last two cases enumeration and importance sampling are impractically slow. Following Booth and Butler (1999), a 5% relative error criterion was used to assess convergence, where the relative error is defined as the 95% margin of error divided by the P-value estimate. This criterion is perhaps too stringent for very small P-values, say $p \leq .01$. For such cases a stopping rule based on the absolute error is preferable.

To verify the behavior of the algorithm in repeated applications, multiple runs were attempted for each example. Sample path plots of the P-value estimates by iteration for 10 independent runs of the algorithm for Examples 1, 2 and 3 are given

Table 4.1: A comparison of asymptotic, exact conditional, and MCMC exact conditional p-values for 6 examples.

Example	Data	DF	-2logλ	P_{χ^2}	P_{MCMC}	P
1 Test of independence	Table 3.3	9	15.49	.078	.118	.114
2 Test of quasi-independence	Table 1.1	5	13.55	.019	.022	.023*
3 Test of quasi-symmetry	Table 4.3	3	2.98	.394	.390	.393*
4 Test of uniform association	Table 1.1	15	16.21	.368	.044	.044
5 Poisson Log-linear Model	Table 4.4	40	50.27	.128	.202	
6 Capture/recapture	Table 4.5	55	57.29	.366	.368	

An * indicates P is a Monte Carlo approximation from another algorithm (estimated to 6 decimal places at the 99% confidence level).

in Figure 4.1. Our complete knowledge of the reference set in Example 4[1] allows us to assess convergence to stationarity by plotting the empirical survival function of the deviance (based on the first 1000 iterations) to the exact (stationary) survival function (Figure 4.2). The vertical line represents the observed deviance for this example. Notice the empirical survival function is nearly identical to the exact.

Two methods for estimating Monte Carlo error were investigated. The first used regeneration (see Section 2.5), which uses the fact that sample averages from tours of the chain between returns to a specific table are independent. We found that the chain returned to tables close to the fitted values often for problems based on smaller contingency tables. However, for larger tables the chain did not regenerate frequently enough to assess convergence efficiently. This method does not seem reasonable for many of the instances when this algorithm would be needed. The second method was the batch means method Geyer (1992) in which averages from non-overlapping sections of the chain are assumed to approximately uncorrelated. The estimate of the Monte Carlo standard error is then the sample standard error of the mean of the averages. To evaluate this approach we compared an empirical estimate of the Monte Carlo variance of the P-value estimate based

[1] Enumerating the reference set took in excess of two months of computing!

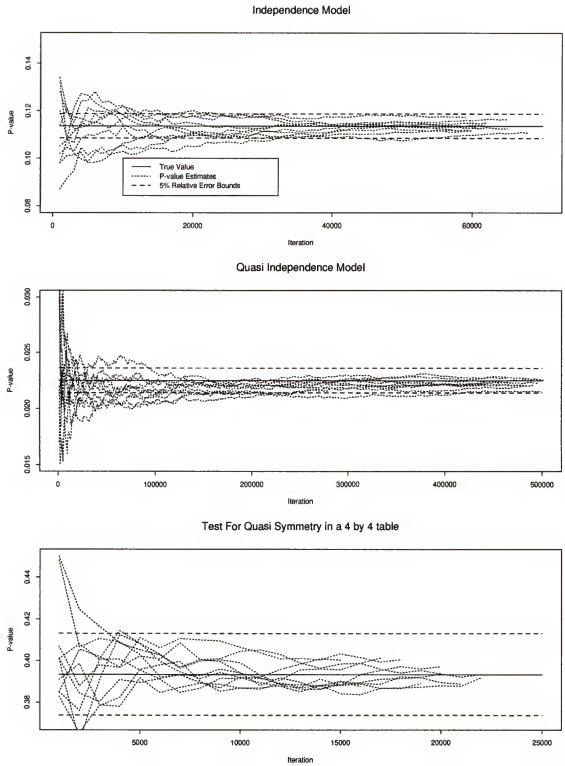

Figure 4.1: Sample path plots of MCMC p-values and 5% relative error bounds for 3 examples.

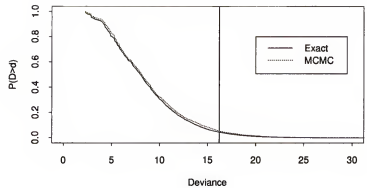

Figure 4.2: MCMC empirical and exact survival function of the deviance for the uniform association model for the pathologist agreement data. The vertical line is the observed deviance.

Table 4.2: A comparison of MCMC variance estimates from batch means and independent runs.

Example	Batch Size	Batch Estimate	Empirical	Example	Batch Size	Batch Estimate	Empirical
1	100	2.3×10^{-5}	2.1×10^{-5}	4	250	3.0×10^{-5}	3.6×10^{-5}
2	100	4.8×10^{-6}	5.4×10^{-6}	5	250	3.9×10^{-4}	3.4×10^{-4}
3	100	4.6×10^{-5}	5.3×10^{-5}	6	500	9.4×10^{-3}	8.1×10^{-3}

on 20 runs of chains of length 20,000 to the estimates obtained by batching. The results of this analysis are given in Table 4.2.

In Examples 1-3 the batch variance estimate was of the same order of magnitude as the empirical estimate as long as the batch size was over 100. However, in general, the batch sizes needed to be increased substantially when the autocorrelations in the chain were larger. As stated earlier, the lower the average number of updated table entries per iteration, the greater the dependency in the chain. Recall that the number of table entries updated at each stage of the Markov Chain is generated from a binomial distribution in our MCMC algorithm. Figure 4.3 shows the autocorrelations using various binomial success probabilities in Example 4. In

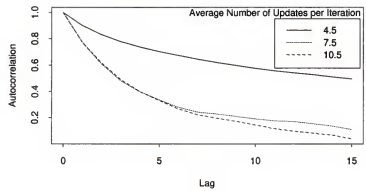

Figure 4.3: Autocorrelations for various update schemes for the uniform association model on the pathologists agreement data.

Table 4.3: Quasi-symmetry data. Cross-classification of family residences in 1980 and 1985.

Residence	Residence in 1985			
in 1980	Northeast	Midwest	South	West
Northeast	11,607	100	366	124
Midwest	87	13,677	515	302
South	172	225	17,819	270
West	63	176	286	10,192

Source: Agresti (1990)

Example 6, in order to obtain a reasonable number of tables with the correct sufficient statistics very few table entries could be updated per iteration. The resulting large autocorrelations induced by this required batches of size 500 to achieve good accuracy. For this example less than .1% of the pseudo samples generated by importance sampling (using the t instrumental distribution) had the correct sufficient statistic, as opposed to 40% with the Metropolis algorithm.

4.6 Discussion

The algorithm discussed in this chapter extends the usefulness of Booth and Butler (1999)'s candidate density to a much wider range of data sets and models. The set of models and data sets that can be analyzed by this MCMC method

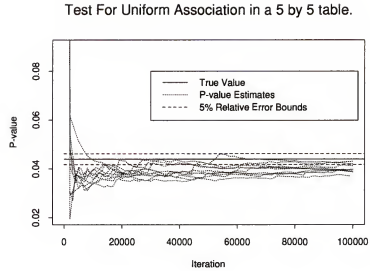

Figure 4.4: An illustration of false convergence.

Table 4.4: Alligators' primary food choice classified by lake, gender and size.

			Primary Food Choice				
Lake	Gender	Size	Fish	Invert	Reptile	Bird	Other
1	Male	Small	7	1	0	0	5
	Male	Large	4	0	0	1	2
	Female	Small	16	3	2	2	3
	Female	Large	3	0	1	2	3
2	Male	Small	2	2	0	0	1
	Male	Large	13	7	6	0	0
	Female	Small	3	9	1	0	2
	Female	Large	0	1	0	1	0
3	Male	Small	3	7	1	0	1
	Male	Large	8	6	6	3	5
	Female	Small	2	4	1	1	4
	Female	Large	0	1	0	0	0
4	Male	Small	13	10	0	2	2
	Male	Large	9	0	0	1	2
	Female	Small	3	9	1	0	1
	Female	Large	8	1	0	0	1

Source: Agresti (1990)
Model (FG, FL, FS, LGS) where F=food choice, L=lake, S=size, G=gender.

Table 4.5: Snowshoe hare data. Classification of capture (1) or not (0) at six separate times.

Capture	Capture 3, 2, 1							
6, 5 , 4	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0	-	3	6	0	5	1	0	0
0 0 1	3	2	3	0	0	1	0	0
0 1 0	4	2	3	1	0	1	0	0
0 1 1	1	0	0	0	0	0	0	0
1 0 0	4	1	1	1	2	0	2	0
1 0 1	4	0	3	0	1	0	2	0
1 1 0	2	0	1	0	1	0	1	0
1 1 1	1	1	1	0	0	0	1	2

Source Coull and Agresti (1999)

Model

$$\log(\mu_{\mathbf{i}}) = \beta_0 + \beta_1 I(i_1 = 1) + \ldots + \beta_6 I(i_6 = 1) + \binom{\sum_{j=1}^{6} i_j}{2} \beta_7$$

where $\mathbf{i} = (i_1, \ldots, i_6)$ is a possible sequence of responses on the 6 capture experiments ((0,0,0,0,0,0), (0,0,1,0,0,0),...) and $\mu_{\mathbf{i}}$ is the Poisson expected mean cell count for response $\mathbf{i}$. A structural zero occurs at (0,0,0,0,0,0) as a hare cannot be in the study if it was never captured.

contains the set that may be analyzed by importance sampling. However, when it works, importance sampling is generally faster than MCMC methods. Thus an efficient adaptive method would first attempt importance sampling then switch to a Markov Chain algorithm if too few valid tables were being generated or if there were numerical problems with the importance weights. There are, of course, tables for which neither method will converge in a reasonable amount of time. For example, high dimensional tables with sparse entries. Another possible use of the normal candidate transition kernel is in conjunction with other (perhaps faster) algorithms to improve their properties in a manner suggested by Patel et al. (1999). For example, this algorithm could be used to convert a reducible chain to an irreducible chain by choosing the transition kernel to be a mixture of the kernels from the reducible chain and our Metropolis based chain.

We conclude with a word of caution concerning Monte Carlo approximations. Figure 4.4 is a plot of 10 runs of a Metropolis algorithm for the uniform association

model that updates the entire table at each iteration. The use of a normal candidate made certain tables practically inaccessible (as discussed earlier) despite the theoretical validity of the normal candidate. The chains are clearly not converging to the true value, however they do appear to be converging. The P-value reported by Booth and Butler (1999) is incorrect essentially for this reason. Replacing the normal update with a t alleviated this problem and offered an improvement that would have otherwise gone unnoticed for both the algorithm presented in this paper and Booth and Butler's importance sampling algorithm had the exact value not been calculated.

CHAPTER 5
ESUP ACCEPT/REJECT SAMPLING

5.1 Introduction

Recall from Section 2.4 accept/reject sampling is a way to generate a random sample from a target density f from which it is difficult to simulate, using a random sample from a more tractable candidate density g. Let F and G denote the corresponding distribution functions. A key requirement for accept/reject sampling is is the existence and discovery of the finite supremum $C = \sup_x f(x)/g(x)$. Let C_{UB} be an upper bound for C. Then the usual accept/reject sampling algorithm is given by Algorithm 2.1.

The value of C_{UB} is the average number of candidate variates required to obtain one target variate (Robert and Casella; 1999, Problem 2.36). The ratio C/C_{UB} measures the efficiency of the upper bound relative to the exact supremum. Although use of an upper bound in place of the exact value produces a random sample of target variables, the bound must be reasonably tight in practice for the method to be efficient. If C_{UB} is replaced with a lower bound, say $C_{LB} < C$, rejection sampling yields variables from a distribution other than the target. Tierney (1994) shows that the density for this distribution is proportional to $\min(f, C_{LB}g)$.

Often in practice C is unknown and it is hard to compute an upper bound that gives an efficient algorithm. In such cases we propose estimating C using a sequence of lower bounds given by $\hat{C} = \max f(X_j)/g(X_j)$, the maximum ratio obtained from the simulated candidate variables. This maximum, or empirical supremum (ESUP), is typically an extremely efficient estimator of C if $C < \infty$

(see Section 5.3). We exploit its fast rate of convergence to show that accept/reject sampling using $\hat{C}$ accepts essentially the same candidates as accept/reject sampling with a known supremum (KSUP accept/reject sampling). In fact, when f and g have finite support, the sequences of accepted values from the ESUP and KSUP algorithms only differ for finitely many repetitions of the algorithm, with probability one. Such a strong result does not quite hold in continuous cases. In that case, we relate the rate of convergence of the empirical supremum to the rate of convergence of the difference between averages computed using the two sequences. Specifically, we show that the ESUP sequence inherits the strong law of large numbers and central limit theorem obeyed by the KSUP sequence. Thus, if the goal is to evaluate an expectation with respect to F, variates generated using the ESUP algorithm may be treated as if they are a random sample from F.

An important and useful property of accept/reject sampling is that f and g need only be known up to normalizing constants. That is, if $f = af^*$ and $g = bg^*$, where a and b are normalizing constants, then step 2 of Algorithm 2.1 is equivalent to "Accept X if $U \leq f^*(X)/C^*g^*(X)$", where $C^* = Cb/a = \sup_x f^*(x)/g^*(x)$. Hence, in most implementations of accept/reject sampling, the exact value of C is not calculated. The ESUP algorithm also has this property.

As $\hat{C}$ depends on the previous candidates, the accepted candidates are dependent. By introducing this dependency, this work differs from two other variations on accept/reject sampling, namely adaptively improving the candidate as the algorithm progresses (Gilks and Wild; 1992; Wild and Gilks; 1993); and recycling the information contained in the uniforms from step 2 of Algorithm 2.1 (Casella and Robert; 1996, 1998). A third more closely related variation adjusts for the use of a possibly incorrect value of C with the independence Metropolis algorithm (Tierney; 1994). We compare this approach with ESUP accept/reject sampling in the discussion at the end of this chapter.

The rest of the chapter is organized as follows. In Section 5.2 we formally develop ESUP and KSUP accept/reject sampling. We then use this development in Section 5.3 to state and prove the main theorems. In Section 5.4 we discuss implementation and the possibility of improving on estimates of C. In Section 5.5 we explore the possibility of diagnosing when $C = \infty$. In Section 5.6 we present several examples. Finally Section 5.7 offers discussion and extensions.

5.2 ESUP Accept/Reject Sampling

Monte Carlo simulation is often useful when a joint density factors as $f(z, y) = h(z \mid y)\pi(y)$, where the conditional and marginal densities, h and π, are known up to normalizing constants. Denote the posterior distribution of Y given $Z = z$ by $f(y|z)$. Then KSUP accept/reject sampling from $f(y|z)$ requires the calculation of

$$C \propto \sup_y h(z \mid y)\frac{\pi(y)}{g(y)} \tag{5.1}$$

(where here $\propto$ signifies that a multiplicative constant was omitted) In particular, taking $g(y) = \pi(y)$ gives $C \propto h(z \mid \hat{y})$ where $\hat{y} = \text{argmax}_y h(z \mid y)$. For example, suppose $h(z \mid y)$ is binomial with n trials and success probability $p(y) = (1+e^{-y})^{-1}$. Let $g(y)$ and $\pi(y)$ be Normal(α, σ^2). Then

$$C = (z/n)^z(1 - z/n)^{n-z}K^{-1},$$

where K is a constant depending on n and z. Notice that if $n \to \infty$ in such a way that z/n remains constant then $C \to \infty$. To prove this we need to show that $K \to 0$ at a rate faster than $(z/n)^z(1 - z/n)^{n-z}$. Up to constants independent of z

and n,

$$\begin{aligned} K &= \int_{\mathbb{R}} p(y)^z (1-p(y))^{n-z} \phi\left(\frac{y-\alpha}{\sigma}\right) dy \\ &\leq \int_{\mathbb{R}} p(y)^z (1-p(y))^{n-z} \phi(0) dy \\ &\doteq \Gamma(z)\Gamma(n-z)/\Gamma(n) \\ &\doteq (z/n)^z (1-z/n)^{n-z}/n \end{aligned}$$

using Stirling's approximation (where here ϕ is the standard normal density).

The choice of candidate here is motivated by the ease with which C could be calculated. However since C can be arbitrarily large, this candidate could be a very bad choice. A more efficient candidate would be allowed to depend on z and n. For example, the Laplace approximation (Section 2.2) shows that $f(y|z)$ is approximately normal with mean equal to the posterior mode and variance equal to the negative inverse of the curvature of the log of the posterior at the mode. Therefore, shifting and scaling a heavy tailed distribution by this mean and standard deviation should produce a candidate that improves with n. However, evaluation of C with g as a shifted and scaled t distribution, for example, is very difficult. This example is considered in more detail in Section 5.6.

ESUP accept/reject sampling simply estimates C with the largest observed value of $f(X)/g(X)$ from the candidate variates X seen as the algorithm progresses. If $\hat{C}$ denotes the current estimate of C, the ESUP algorithm is:

Algorithm 5.1 ESUP Accept/Reject Sampling

1. Generate $X \sim G$ and $U \sim (0,1)$ independently.
2. Accept X if $U \leq f(X)/\hat{C}g(X)$.
3. Update $\hat{C} = \max\{\hat{C}, f(X)/g(X)\}$
4. Return to step 1.

Like KSUP accept/reject sampling, this algorithm does not require the normalizing constants for f and g. Moreover, since $\hat{C} \to C$ with probability 1, if

both algorithms are run with the same candidates and uniforms, the probability that the ESUP algorithm accepts candidates that the KSUP algorithm rejects goes to 0 with the number of iterations. On the other hand, since $\hat{C} \leq C$, the probability that the ESUP algorithm rejects a candidate that the KSUP algorithm accepts is always zero.

For a theoretical discussion of both algorithms, we need some additional notation. Let $\{X_{ij}\}_{ij \in \mathbb{N} \times \mathbb{N}}$ be a doubly-indexed sequence of independent variates from G, mutually independent of the doubly-indexed sequence $\{U_{ij}\}_{ij \in \mathbb{N} \times \mathbb{N}}$ of independent uniform$(0,1)$ variates. The subscript identifies the j^{th} candidate and uniform variates used in generating the i^{th} observation from the target distribution. Let

$$\tau_i = \min \left\{ j \in \mathbb{N} \middle| U_{ij} \leq \frac{f(X_{ij})}{Cg(X_{ij})} \right\}$$

and define $Y_i = X_{i\tau_i}$. Then the sequence $\{Y_i\}$ is generated according to Algorithm 2.1. We refer to τ_i as the i^{th} (KSUP) acceptance number. Evidently the $\{\tau_i\}$ are independent geometric random variables with success probabilities C^{-1}.

We use similar notation to formalize Algorithm 5.2, but distinguish the acceptance number and accepted candidate with a tilde. Thus, the i^{th} acceptance number from the ESUP algorithm is

$$\tilde{\tau}_i = \min \left\{ j \in \mathbb{N} \middle| U_{ij} \leq \frac{f(X_{ij})}{\hat{C}_i g(X_{ij})} \right\},$$

and the i^{th} accepted candidate is $\tilde{Y}_i = X_{i\tilde{\tau}_i}$, where the empirical sup is defined recursively by

$$\hat{C}_{i+1} = \max \left\{ \frac{f(X_{i1})}{g(X_{i1})}, \hat{C}_i \right\}. \tag{5.2}$$

In these formal descriptions of the algorithms we update (5.2) only once for every *accepted* candidate, in contrast to Algorithm 5.2 where updating occurs for *every* candidate. This simplifies both notation and theory, as $\hat{C}_i$ is a largest order

statistic based on a fixed number $i-1$ of observations, rather than a random number of them. Also, $\hat{C}_i$ defined in this way is independent of X_{ij} for all j. In Section 5.3 we argue that our main results continue to hold with any scheme that implements larger lower bounds than those defined in (5.2).

The recursive definition of $\hat{C}_i$ requires an initial $\hat{C}_1$. As $C \geq 1$, we set $\hat{C}_1 = 1$ in the next section. In practice, where $\hat{C}_i$ is only evaluated up to a constant of proportionality one might set $\hat{C}_1 = 0$.

The three main assumptions needed for ESUP accept/reject sampling are

A1. $\mathcal{X}_F \subset \mathcal{X}_G$,

A2. $C \equiv \sup \left\{ \frac{f(x)}{g(x)} \,|\, x \in \mathcal{X}_F \right\} < \infty$,

A3. $C = f(x_C)/g(x_C)$ for some $x_C \in \mathcal{X}_F$,

where $\mathcal{X}_F$ and $\mathcal{X}_G$ denote the supports of F and G respectively. Assumptions A1 and A2 are required for KSUP rejection sampling while assumption A3 is not. In most situations all three can be satisfied by choosing a candidate with heavier tails than the target. Specifically, they hold if f and g are bounded and g dominates f outside a compact subset of $\mathcal{X}_F$.

5.3 Convergence

The key quantity for comparing ESUP and KSUP sequences is $P(Y_i \neq \tilde{Y}_i)$, the probability the ESUP algorithm erroneously accepts a candidate that the KSUP algorithm rejects. In the discrete case, $\sum_i P(Y_i \neq \tilde{Y}_i) < \infty$. That is, with probability one, the output from the two algorithms differs for only finitely many i. Therefore the ESUP sequence has all of the limiting properties of the independent identically distributed sequence from the KSUP algorithm.

Theorem 5.1. *If the support of F is countable then*

$$P(Y_i \neq \tilde{Y}_i \text{ infinitely often in } i) = 0.$$

Proof. Assumption A3 gives $C = f(x_C)/g(x_C)$ for some $x_C \in \mathcal{X}_F$. If $\gamma = \min\{i \in \mathbb{N} | X_{i1} = x_C\}$, then γ is geometric with success probability $g(x_C)$, where $g(x_C) > 0$ by the assumption that $\mathcal{X}_F \subset \mathcal{X}_G$. As the algorithms are identical when $\hat{C}_i = C$, it follows that $[Y_i \neq \tilde{Y}_i] \subset [\gamma \geq i]$. Thus

$$P(Y_i \neq \tilde{Y}_i) \leq P(\gamma \geq i) = \{1 - g(x_C)\}^{i-1}$$

and hence $\sum_i P(Y_i \neq \tilde{Y}_i) < \infty$. □

In practice Theorem 5.1 applies also to continuous cases, because C is typically evaluated only up to a given (computer) accuracy. For a fair comparison of KSUP and ESUP algorithms we should then use $\hat{C}_i$ to estimate C to within the same tolerance, and with probability one this must occur within finitely many iterations. More formally, however, in many continuous settings $\sum_i P(Y_i \neq \tilde{Y}_i)$ may not be finite because $P(Y_i \neq \tilde{Y}_i)$ is $\mathcal{O}(i^{-1})$. The following three lemmas establish this rate of convergence, which is needed to prove our main result, stated in Theorem 5.2. These arguments only require assumptions A1–A3. In Section 5.4, under the assumption that $\log(f/g)$ is smooth and unimodal, we argue $P(Y_i \neq \tilde{Y}_i) = \mathcal{O}(i^{-2})$.

Lemma 5.1. *Under assumptions A1 and A2*

$$P(Y_i \neq \tilde{Y}_i) \leq E(C/\hat{C}_i) - 1,$$

the $\mathcal{L}_1$ *distance* $E\left(\left|\frac{C}{\hat{C}_i} - 1\right|\right)$ *between* $C/\hat{C}_i$ *and 1.*

Proof. That $P(Y_i \neq \tilde{Y}_i) \leq P(\tau_i \leq \tilde{\tau}_i)$ is clear as $Y_i \neq \tilde{Y}_i$ is equivalent to $X_{i\tau_i} \neq X_{i\tilde{\tau}_i}$. To prove $P(\tau_i \leq \tilde{\tau}_i) \leq E(C/\hat{C}_i) - 1$, consider the events

$$A_{ij} = \left\{U_{ij} \leq \frac{f(X_{ij})}{g(X_{ij})C}\right\}, \quad B_{ij} = \left\{U_{ij} \leq \frac{f(X_{ij})}{g(X_{ij})\hat{C}_i}\right\}.$$

As $\hat{C}_i < C$ it follows that $A_{ij} \subset B_{ij}$. The probabilities of A_{ij} and B_{ij} are

$$\begin{aligned} P(A_{ij}) &= 1/C, \\ P(B_{ij}) &= E\left[P\left(U_{ij} \leq \frac{f(X_{ij})}{g(X_{ij})\hat{C}_i}\middle| X_{ij}, \hat{C}_i\right)\right] \\ &= E\left[\min\left(\frac{f(X_{ij})}{g(X_{ij})\hat{C}_i}, 1\right)\right]. \end{aligned}$$

As $\hat{C}_i$ is independent of X_{ij} for all j, and the X_{ij} are independent and identically distributed, $P(B_{ij})$ is constant for all j.

The probability that the ESUP accept/reject sampler accepts $X_{ij'}$ while the KSUP accept/reject sampler accepts X_{ij} $(for j' < j)$ is the probability that the first $j'-1$ candidates are rejected by both algorithms, times the probability that $X_{ij'}$ is accepted by the ESUP algorithm but not the KSUP algorithm, times the probability the next $j-j'$ candidates are rejected by the KSUP algorithm, times the probability that X_{ij} is accepted by the KSUP algorithm. Equivalently, we have

$$\begin{aligned} P(\tilde{\tau}_i = j', \tau_i = j) &= \left[\prod_{l=1}^{j'-1} 1 - P(B_{il})\right] \left[P(B_{ij'} \cap A_{ij'}^c)\right] \left[\prod_{l=j'+1}^{j-1} 1 - P(A_{il})\right] P(A_{ij}) \\ &\leq \left[\prod_{l=1}^{j'-1} 1 - P(A_{il})\right] \left[P(B_{ij'}) - P(A_{ij'})\right] \left[\prod_{l=j'+1}^{j-1} 1 - P(A_{il})\right] P(A_{ij}) \\ &= \left[P(B_{i1}) - \frac{1}{C}\right] \left(1 - \frac{1}{C}\right)^{j-2} \frac{1}{C}. \end{aligned}$$

Summing over $j' < j$ gives $P(\tilde{\tau}_i < j, \tau_i = j) \leq (j-1)\left[P(B_{i1}) - \frac{1}{C}\right]\left(1 - \frac{1}{C}\right)^{j-2} \frac{1}{C}$, so

$$\begin{aligned} P(\tilde{\tau}_i < \tau_i) &\leq \sum_{j=1}^{\infty} P(\tilde{\tau}_i < j, \tau_i = j) \\ &= \left[P(B_{i1}) - \frac{1}{C}\right] \sum_{j=1}^{\infty} (j-1) \left(1 - \frac{1}{C}\right)^{j-2} \frac{1}{C} \\ &= \left[P(B_{i1}) - \frac{1}{C}\right] \left(1 - \frac{1}{C}\right)^{-1} (C-1) = \left[P(B_{i1}) - \frac{1}{C}\right] C. \end{aligned}$$

To complete the proof, we must show $P(B_{i1}) \leq E[1/\hat{C}_i]$. But

$$\begin{aligned} P(B_{i1}) &= E\left[\min\left(\frac{f(X_{i1})}{g(X_{i1})\hat{C}_i}, 1\right)\right] \\ &\leq E\left[\frac{f(X_{i1})}{g(X_{i1})\hat{C}_i}\right] \\ &= E\left[\frac{f(X_{i1})}{g(X_{i1})}\right] E\left[\frac{1}{\hat{C}_i}\right] \\ &= E\left[\frac{1}{\hat{C}_i}\right]. \end{aligned}$$

□

Lemma 5.2. *Let $\{Z_k\}$ be a sequence of independent random variables from a continuous density w, with associated distribution function W. Suppose that for some b, $W(b) = 1$ and $W(b-\epsilon) < 1$ for all $\epsilon > 0$, where $w(b) > 0$. Let $Z_{(i)} = \max\{Z_k | k = 1, \ldots, i\}$. Then $b - E(Z_{(i)}) = \mathcal{O}(i^{-1})$. Moreover, the same rate holds for sample minima with finite lower bounds.*

Proof. Suppose that $b = 1$ and $Z_k > 0$. Then

$$\begin{aligned} 1 - E\left[Z_{(i)}\right] &= 1 - \int_0^1 \left(1 - W^i(t)\right) dt \\ &= \int_0^1 W^i(t) dt \\ &= \int_0^{1-\epsilon} W^i(t) dt + \int_{1-\epsilon}^1 W^i(t) dt \\ &\leq W^i(1-\epsilon)(1-\epsilon) + \int_{1-\epsilon}^1 W^i(t) dt. \end{aligned}$$

As $W^i(1-\epsilon)(1-\epsilon) < \mathcal{O}(i^{-1})$, we need only investigate $\int_{1-\epsilon}^1 W^i(t) dt$.

By assumption w is continuous from the left and $w(1) > 0$. Let k and ϵ be positive real numbers such that $w(x) > k$ for $1 - \epsilon \leq x \leq 1$ and choose $0 < p < 1$ so that $k > p(1-\epsilon)^{-1}$. Then

$$w(x) > k > p(1-\epsilon)^{-1} \geq p(1-\epsilon)^{p-1} \geq px^{p-1}, \quad 1 - \epsilon \leq x \leq 1.$$

Hence

$$\int_t^1 w(x)dx \geq \int_t^1 px^{p-1}dx$$

which implies $W(1) - W(t) \geq 1 - t^p$. Thus, for $1 - \epsilon \leq t \leq 1$, we have $W^i(t) \leq t^{ip}$. The result then follows from the fact that

$$\int_{1-\epsilon}^1 t^{ip}dt = \frac{1}{ip+1} - \frac{(1-\epsilon)^{ip+1}}{ip+1} = \mathcal{O}(i^{-1}).$$

When the Z_i might be negative but have upper bound 1, let $Z'_{(i)} = \max(Z_{(i)}, 0)$. As $Z'_{(i)}$ is the maximum of non-negative random variables and previous paragraph shows its rate of convergence is $\mathcal{O}(i^{-1})$. Then $Z_{(i)}$ is tail equivalent to $Z'_{(i)}$, yielding the result for all independent identically distributed sequences bounded by 1.

Now assume b is not necessarily 1. Then

$$Z_k = Z_k - b + 1 + (b-1) = Z_k^* + (b-1).$$

Hence the Z_k^* have upper bound 1 and hence

$$\mathcal{O}(i^{-1}) = 1 - E[Z_{(i)}^*] = b - E[Z_{(i)}].$$

Finally, noting that the sample minimum is simply the negative of a sample maximum, we have the corresponding results for minima. □

Lemma 5.3. *Under assumptions A1–A3,* $P(Y_i \neq \tilde{Y}_i) = \mathcal{O}(i^{-1})$.

Proof. By Lemma 5.1 we need only show $E(C/\hat{C}_i) - 1 = \mathcal{O}(i^{-1})$. Note that $C/\hat{C}_i = \min\{Cg(X_{k1})/f(X_{k1}) | k = 1, \ldots, i-1\}$. That is, $C/\hat{C}_i$ is the minimum of $i-1$ independent random variables bounded from below at 1. In light of Lemma 5.2 we need only show that the density of the random variable $Z_k = Cg(X_{k1})/f(X_{k1})$ is strictly positive at 1. But, the density of Z_k is strictly positive for every value $z = Cg(x)/f(x)$ for which $g(x) > 0$. In particular, by assumption A3, the density of Z_k is strictly positive at $1 = Cg(x_C)/f(x_C)$. □

We can now prove our main result. Let h be a real-valued, Borel measurable function. Let $\mu_h = E\{h(Y_i)\}$ and suppose that $\sigma_h^2 = \mathrm{Var}\{h(Y_i)\} < \infty$. Let $h_n = n^{-1}\sum_{i=1}^n h(Y_i)$ denote the sample average from the KSUP chain and let $\tilde{h}_n$ denote the sample average from the corresponding ESUP chain. Almost sure convergence of $\sqrt{n}(h_n - \tilde{h}_n)$ to zero is sufficient to prove that $\{\tilde{h}_n\}$ inherits the strong law of large numbers and central limit theorem obeyed by $\{h_n\}$.

Theorem 5.2. *If $E\{h(Y_i)^\delta\} < \infty$ for some $\delta > 2$ and assumptions A1–A3 hold then*

$$\tilde{h}_n \xrightarrow{a.s.} \mu_h, \qquad \sqrt{n}(\tilde{h}_n - \mu_h) \xrightarrow{\mathcal{D}} N(0, \sigma_h^2).$$

Proof. We prove the SLLN and CLT simultaneously by showing that $\sqrt{n}(h_n - \tilde{h}_n)$ converges almost surely to zero. By Lemma 5.3, $P(Y_i \neq \tilde{Y}_i) = \mathcal{O}(i^{-1})$, so $\sum_{i=1}^\infty P(Y_i \neq \tilde{Y}_i)\epsilon/\sqrt{i} < \infty$ for any $\epsilon > 1/2$. If we let $\epsilon = (\delta - 1)/\delta$, then $\epsilon > 1/2$. Assume for now that $E\left[\left\{h(Y_i) - h(\tilde{Y}_i)\right\}^\delta\right]$ is bounded in i. Then

$$\begin{aligned}
\sum_{i=1}^\infty i^{-1/2} E\left\{\left|h(Y_i) - h(\tilde{Y}_i)\right|\right\} &= \sum_{i=1}^\infty i^{-1/2} E\left\{\left|h(Y_i) - h(\tilde{Y}_i)\right| I_{\{Y_i \neq \tilde{Y}_i\}}\right\} \\
&\leq \sum_{i=1}^\infty i^{-1/2}\left(E\left[\left\{h(Y_i) - h(\tilde{Y}_i)\right\}^\delta\right]\right)^{1-\epsilon} P\left(Y_i \neq \tilde{Y}_i\right)^\epsilon \\
&< \infty,
\end{aligned}$$

where the inequality follows by the Hölder inequality. Thus, by the monotone convergence theorem

$$E\left\{\sum_{i=1}^\infty i^{-1/2}\left|h(Y_i) - h(\tilde{Y}_i)\right|\right\} < \infty,$$

and hence $\sum_{i=1}^\infty i^{-1/2}\left|h(Y_i) - h(\tilde{Y}_i)\right| < \infty$ almost surely. The result then follows by Kronecker's lemma (Chow and Teicher; 1997, page 114).

As $E\{h(Y_i)^\delta\}$ is constant, to show that $E\left[\left\{h(Y_i) - h(\tilde{Y}_i)\right\}^\delta\right]$ is bounded we need only show $E\{h(\tilde{Y}_i)^\delta\}$ is bounded in i. Conditional on $\tilde{C}_i$, the density of $\tilde{Y}_i$ is

proportional to $\min(f, \hat{C}_i g)$ (Tierney; 1994). Therefore

$$\begin{aligned} E\left\{h(\tilde{Y}_i)^\delta \mid \hat{C}_i\right\} &= \int h(x)^\delta \min\{f(x), \hat{C}_i g(x)\}\, dx \Big/ \int \min\{f(x), \hat{C}_i g(x)\}\, dx \\ &\leq \int h(x)^\delta f(x)\, dx \Big/ \int \min\{f(x), \hat{C}_i g(x)\}\, dx \\ &= E\left\{h(Y_1)^\delta\right\} \Big/ \int \min\{f(x), \hat{C}_i g(x)\}\, dx \\ &\leq E\left\{h(Y_1)^\delta\right\} \Big/ \int \min\{f(x), g(x)\}\, dx, \end{aligned}$$

since $\hat{C}_i \geq 1$. Taking expectations over the distribution of $\hat{C}_i$ yields the result. □

The proof of Theorem 5.2 shows that the necessary rate of convergence of $P\left(Y_i \neq \tilde{Y}_i\right)$ decreases as δ increases. For example, when $\delta = 4$ the result requires that rate $P\left(Y_i \neq \tilde{Y}_i\right) = \mathcal{O}(i^{-k})$ for some $k > 2/3$. That is, in most situations the i^{-1} rate is faster than that required for the theorem to hold.

We end this section with a discussion of better estimates of C. Let $\{\hat{Y}_i\}$ be a sequence of accepted values from an ESUP accept/reject sampler implementing larger lower bounds for C than $\hat{C}_i$. Notice that $\hat{Y}_i$ satisfies $P(Y_i \neq \hat{Y}_i) \leq P(Y_i \neq \tilde{Y}_i)$. The crucial quantity in Theorems 5.1 and 5.2 is the rate of convergence of $P(Y_i \neq \tilde{Y}_i)$. As this rate dominates $P(Y_i \neq \hat{Y}_i)$, the sequence $\{\hat{Y}_i\}$ inherits the same properties as $\tilde{Y}_i$. Specifically, this is the case if the maximum is updated with every candidate rather than once for every accepted candidate.

5.4 Confidence bounds

Under the assumption that $\log(f/g)$ is smooth and unimodal at x_C, a more precise description of the asymptotic behavior of $\hat{C}$ is possible. Specifically, the rate of convergence of $\hat{C}$ to C is faster than derived in Section 5.3. Also we can derive an upper confidence bound for C, which results in an ESUP algorithm that produces exact i.i.d. samples with a high probability.

Let $V_i = v(X_{j1}) = \log\{f(X_{j1})/g(X_{j1})\}$, where, as before, the X_{j1} are independent variables from G. Note that $\hat{C}_{i+1} = \exp(\max V_i)$ and $C = \exp\{v(x_C)\}$. In many cases the function $v(x)$ will be smooth and unimodal, having a maximum at x_C, near which it is concave. Then, for X_{j1} close to x_C, we have

$$V_j \doteq v(x_C) + \frac{1}{2}v''(x_C)(X_{j1} - x_C)^2.$$

Thus, as $v(x_C) = \log C$ and $v''(x_C) < 0$, for u close to (but less than) $\log C$,

$$\begin{aligned} P(V_j > u) &\doteq P\left\{\log C + \frac{1}{2}v''(x_C)(X_{j1} - x_C)^2 > u\right\} \\ &= P\left(|X_{j1} - x_C| < [2(\log C - u)/\{-v''(x_C)\}]^{1/2}\right) \\ &\doteq 2\,[2(\log C - u)/\{-v''(x_C)\}]^{1/2}\, g(x_C), \end{aligned}$$

where the final approximation follows by the mean value theorem (for integrals). Therefore, $P(V_j > u)$ can be approximated by $a'(\log C - u)^{1/2}$, say. Then, for u less than but close to $\log C$,

$$\begin{aligned} P\left\{\log \hat{C}_{i+1} \le u\right\} &\doteq \left\{1 - a'(\log C - u)^{1/2}\right\}^i \\ &\doteq \exp\left\{-ia'(\log C - u)^{1/2}\right\}, \end{aligned}$$

where recall $\log \hat{C}_{i+1} = \max(V_1, \ldots, V_i)$. Therefore, as $i \to \infty$, $i^2\{\log C - \log \hat{C}_i\}$ follows the Weibull distribution with shape $1/2$, unknown scale parameter and unknown endpoint $\log C$. Smith (1985, Section 5) argues that in such cases the endpoint can be efficiently estimated by $\log \hat{C}_i$. Further notice that the Weibull approximation suggest

$$E\left[\log\left(\frac{C}{\hat{C}_i}\right)\right] \doteq E\left[\frac{C}{\hat{C}_i} - 1\right] = \mathcal{O}(i^{-2}).$$

Thus, applying Lemma 5.1, $P(Y_i \neq \tilde{Y}_i) = \mathcal{O}(i^{-2})$ and hence the $\{Y_i\}$ are tail equivalent to $\{\tilde{Y}_i\}$.

Table 5.1: Average and median efficiencies (%) estimated from 500 replicates of simulating half-normal variables using an exponential candidate and normal variables using a t_3 envelope, for samples of different sizes. The upper number is the median, the lower the average.

	Sample size n								
	2	3	4	5	10	20	30	40	50
Exp/Half-normal	0.0	0.6	4.4	9.4	48	86	93	97	98
	8.0	16	22	27	49	77	87	92	95
t_3/normal	6.8	26	38	47	80	93	97	98	99
	21	35	43	50	72	88	94	96	98

We now extend these results to obtain an upper confidence limit for $\log C$. By the previous work, $-\{\log a' + \frac{1}{2}\log(\log C - V_j)\}$ follows a standard exponential distribution (in the limit). Notice that the Taylor series approximations improve the closer V_j is to $\log C$. This in mind, it is not surprising, that this approximation improves when considering the larger order statistics of the V_j. Thus the difference $\frac{1}{2}\log(\log C - V_{(i-1)}) - \frac{1}{2}\log(\log C - V_{(i)})$ (where $V_{(j)}$ is the jth order statistic of $V_1, \ldots, V_i$) follows an approximate standard exponential distribution, where the approximation is fairly good for large n. Let $e_\alpha = -\log(1-\alpha)$ denote the $1-\alpha$ quantile of the standard exponential distribution. Then

$$\alpha \doteq P\left\{\frac{1}{2}\log\left(\frac{\log C - V_{(i-1)}}{\log C - V_{(i)}}\right) \geq e_\alpha\right\} = P\left\{\log C \leq V_{(i)} + \frac{V_{(i)} - V_{(i-1)}}{\exp(2e_\alpha) - 1}\right\},$$

giving an approximate level α upper confidence bound for $\log C$ and hence for C. This suggests taking $\alpha = 0.05$, say, and using the sequence $\log \hat{C}_{UB} = V_{(i)} + (V_{(i)} - V_{(i-1)})/\{\exp(2e_\alpha) - 1\}$ computed at each iteration to reject or accept X_j.

Table 5.1 shows the efficiencies of this approach for simulating normal data using a t_3 candidate distribution and simulating half-normal data using an exponential candidate distribution. Here the efficiency is defined to be the percentage of times the ESUP algorithm's decision (using the upper confidence limit) to accept or reject a candidate agreed with the KSUP algorithm's decision. Notice, that using

the confidence bound allows for the possibility of the ESUP algorithm rejecting a candidate that the KSUP algorithm accepts. Notice for small sample size n the efficiency can be low, but it rapidly approaches 100% as n grows and $\hat{C}_{UB}$ decreases towards C.

In some cases, the maximum C is attained at two values of x_C; this happens, for example, when a t candidate is used for the normal density. In this case, the properties of the V_j are left unchanged. However, it may occur that $v(x)$ has a maximum at the end of the range of x, with $v'(x_C) \neq 0$ and $V_j = v(x_C) + (X_{j1} - x_C)v'(x_C) + \cdots$, where $v'(x_C) < 0$. A variant of the previous argument then shows that $c''(\log C - V_j)$ has a unit exponential distribution and that the rate of convergence of $\max V_j$ to $\log C$ is no faster than the $\mathcal{O}(i^{-1})$ shown in Section 5.3. However, a similar argument still applies, giving upper α confidence limit $(V_{(i)} - cV_{(i-1)})/(1-c)$, where $c = i^{-1}(i-1)F_{2,2}(\alpha)$, with $F_{2,2}(\alpha)$ the α quantile of the $F_{2,2}$ distribution.

5.5 Diagnostics

The argument of Section 5.4 also provides diagnostics to assess whether a given g provides a suitable candidate for f, if there is doubt. If $C = \sup_x f(x)/g(x) = \infty$, then the upper tail behavior of the random variable V_j will be quite different from that for finite C. Regardless of C, the large sample joint distribution of the V_j exceeding some threshold will be approximately that of independent generalized Pareto variables (Davison and Smith; 1990). This approximation assumes that the number of exceedances is small relative to the overall sample size and holds in wide generality (Pickands; 1975). The generalized Pareto distribution is

$$H(w) = \begin{cases} 1 - (1 + \kappa w/\sigma)_+^{-1/\kappa}, & \kappa \neq 0, \\ 1 - \exp(-w/\sigma), & \kappa = 0, \end{cases} \tag{5.3}$$

where $\sigma > 0$. The parameter κ determines the shape of the upper tail, with $\kappa < 0$ giving a finite upper bound and therefore finite C. This suggest diagnosing an infinite C by testing the null hypothesis $\kappa = 0$ against the alternative $\kappa > 0$[1] The score statistic for $\kappa = 0$ under model (5.3) is equivalent to Greenwood's statistic, $G_s = \sum_{j=1}^{k} S_j^2$ where $S_j = U_j - U_{j-1}$ for $U_1, \ldots, U_{k-1}$ a random sample from the $U(0,1)$ distribution and $U_0 = 0$, $U_k = 1$. The distribution of G_s has been extensively tabulated (Burrows; 1979; Currie; 1981; Stephens; 1981); percentage points are also easily calculated by simulation.

In our context we suggest that the threshold be taken to be $V_{(n-k)}$, for a moderate value of k ($k = 21$, say). The test is then applied to the spacings $S_j = S_j' / \sum_{j=1}^{k} S_j'$ of $S_j' = V_{(j)} - V_{(j-1)}$, for $j = i - k + 1, \ldots, i$. Values of G_s in the upper tail of its null distribution suggest that $\kappa > 0$ and that the candidate density has been badly chosen. We note that this statistic is most useful when it rejects as the null hypothesis, $\kappa = 0$, assumes an infinite value of C.

To test this diagnostic, we simulated exceedances using a normal target and a t_3 candidate distribution ($C < \infty$) and a t_3 target with a normal candidate distribution ($C = \infty$). Figure 5.5 shows the P-value for the score test computed every 100 iterations for i between 1,000 and 10,000 using the largest $k = 21$ exceedances. When $C = \infty$ the P-value tends to remain close to 0. However, when $C < \infty$ the P-value exhibits erratic behavior.

5.6 Examples

In this section we illustrate the algorithm through several examples. First we use simulation studies to explore the convergence of the empirical sup in

[1] The more natural alternative, $\kappa < 0$, does not lead to a easily calculated test statistic.

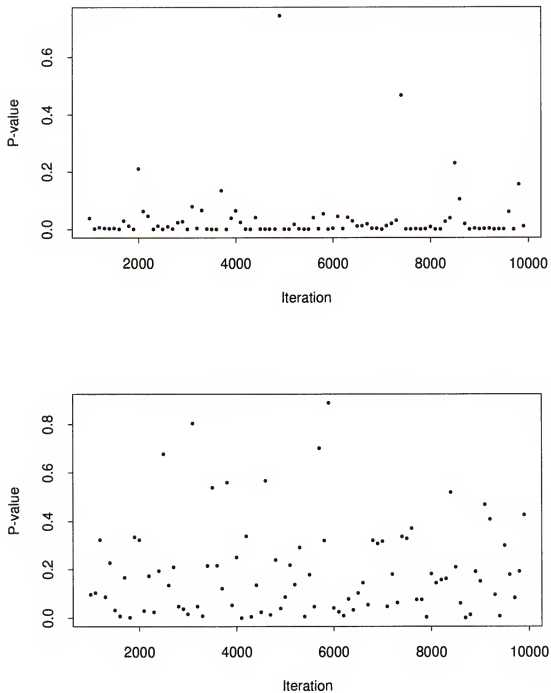

Figure 5.1: Plots of P-value by iteration for generating a normal with a t_3 candidate (above) and a t_3 with a normal candidate (below).

an example similar to the example from Section 5.2. Then we apply the ESUP algorithm to fit the well known "Pump Failure Data" (see, for example, Robert and Casella; 1999), perform an exact conditional test of the quasi-symmetry data in Table 4.3, and estimate the conditional likelihood for binomial response data. Further examples are given in Chapter 6 where ESUP accept/reject sampling is used in the MCEM algorithm.

We first analyze the example from Section 5.2 in detail. The extension of this example to the full MCEM algorithm is considered in Chapter 6. Here, we consider two candidates for a accept/reject sampler used to simulate from the "posterior" of a random intercept logistic/normal model. The random intercept model specifies that given p, Z is binomial(n, p) with $\log\{p/(1-p)\} = Y$, where $Y \sim N(\alpha, \sigma^2)$. Such a model arises when obtaining shrinkage estimates for small area estimation (Agresti et al.; 2000). Also, simulating from the distribution of Y conditional on Z, α and σ allows one to replace intractable integrals with Monte Carlo estimates when finding marginal maximum likelihood estimates via the EM algorithm (Booth and Hobert; 1999). For simplicity we consider simulating from $Y \mid Z = z, \alpha = 1, \sigma = 0.5$.

As stated earlier, accept/reject sampling from $Y \mid Z = z$ using the marginal distribution of Y as the candidate distribution yields $C \propto (z/n)^z(1 - z/n)^{n-z}$. We repeatedly simulated M variates, simultaneously applying KSUP and ESUP accept/reject sampling with the same candidates and uniforms as described in Section 5.3. Table 5.2 gives the average number of times the ESUP algorithm erroneously accepted a candidate in M simulations for various n with $z/n = 1/3$. In the notation of the chapter this average estimates

$$E\left[\sum_{i=1}^{M} I_{\{Y_i \neq \tilde{Y}_i\}}\right] = \sum_{i=1}^{M} P(Y_i \neq \tilde{Y}_i).$$

Table 5.2: Average number of differences (AND) and acceptance rate (AR) for marginal and Laplace candidates with $z/n = 1/3$ for $M = 1,000$.

		Marginal		Laplace/t	
n	z	AND	AR	AND	AR
9	3	20.56	0.11	0.28	0.85
12	4	18.359	0.07	0.43	0.85
15	5	17.03	0.05	0.27	0.85
18	6	16.09	0.04	0.25	0.86
21	7	14.429	0.03	0.31	0.86
24	8	13.703	0.02	0.28	0.86
27	9	13.134	0.02	0.32	0.86
30	10	11.999	0.02	0.25	0.86

The empirical supremum stabilized so fast that on average the accepted variates from the two algorithms differed on less than 21 occasions for all cases considered. Furthermore, the results are exactly the same for $M = 10,000$. Thus, after the empirical supremum stabilizes, the algorithms accept the same candidates. In effect it does not matter which of the two algorithms is used in practice.

The choice of a $N(\alpha, \sigma^2)$ candidate greatly simplified calculation of C. As noted earlier, the acceptance rate for this candidate decreases as n increases. For example, with $n = 9$ the acceptance rate was 11% as opposed to only 2% for $n = 30$. However, we can construct a far more effective candidate by Laplace approximation (Section 2.2). Let $f(z, y; \alpha, \sigma)$ denote the joint density of Z and Y, and let μ be such that

$$\frac{\partial}{\partial y} \log f(z, y; \alpha, \sigma)\Big|_{y=\mu} = 0.$$

Then μ is a second order (Laplace) approximation to $E[Y \mid Z = z, \alpha, \sigma]$. Furthermore,

$$\theta = \left\{ -\frac{\partial^2}{\partial y^2} \log f(z, y; \alpha, \sigma) \right\}^{-1}\Bigg|_{y=\mu}$$

is the corresponding Laplace approximation for $\mathrm{Var}(Y \mid Z = z, \alpha, \sigma)$ (see Booth and Hobert; 1999, for details). Shifting and scaling a distribution by μ and $\theta^{1/2}$

Table 5.3: Number of failures and failure times of 10 pumps.

Pump	1	2	3	4	5	6	7	8	9	10
Failures	5	1	5	14	3	19	1	1	4	33
Time	94.32	15.72	62.88	125.76	5.24	31.44	1.05	.105	2.10	10.48

Source: Tierney (1994)

provides a very accurate candidate as long as assumptions A1–A3 are met. As all three assumptions are met by the normal marginal candidate, we need only choose a candidate whose tails dominate a $N(\alpha, \sigma^2)$ distribution.

Using the t_3 distribution centered at μ and scaled by $\theta^{1/2}$ as a candidate results in an acceptance rate uniformly (empirically calculated to be) higher than 85% for all values of n (see Table 5.2). A closed form expression for C is not available here. Also, since the ratio f/g has multiple extrema (see Figure 5.6) one must be careful when applying a numerical maximization routine. The average number of times the ESUP algorithm differed from the KSUP algorithm for the Laplace/t candidate distribution are given in Table 5.2. For all values of M and n, $\hat{C}$ converged so rapidly that the average number of times the ESUP algorithm erroneously accepted a candidate was less than 1. This example illustrates the usefulness of our ESUP algorithm. A candidate distribution can be chosen without worrying about the numerical maximization necessary for exact accept/reject sampling, with essentially no loss in efficiency, and with diagnostics that warn if the choice is bad.

In our next example we consider the "pump failure" data given in Gaver and O'Muircheartaigh (1987). The data is reproduced in Table 5.3. Here the response, y_i for $i = 1, \ldots, 10$ represents the number of failures at run times t_i. A Hierarchical model for this data (Robert and Casella; 1999; Tierney; 1994) specifies

$$y_i | \lambda_i \sim \text{Poisson}(t_i \lambda_i)$$

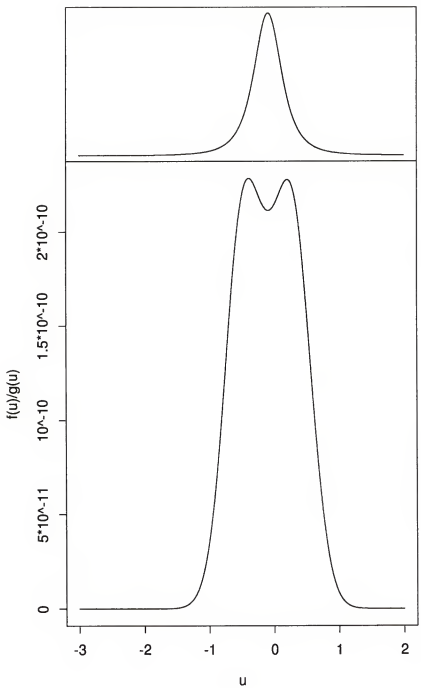

Figure 5.2: Plot of $g(y)$ and $f(y)/g(y)$ by y for $n_i = 30$, $\alpha = 1$, $\sigma = 1/2$.

with

$$\lambda_i|\beta \sim \text{Gamma}(\alpha, \beta)$$

and

$$\beta \sim \text{Gamma}(\gamma, \delta).$$

The models arises from assuming the failure times follow a Poisson process. Jones and Hobert (personal communication) generate from the posterior by sequentially simulating from $f(\beta|\mathbf{y})$ and then from $f(\lambda_i|\beta, \mathbf{y})$ for $i = 1, \ldots, 10$. It is easy to generate from the independent gamma distributions of $\lambda_i|\beta, \mathbf{y}$. However, simulating from $f(\beta|\mathbf{y})$ is more difficult. Jones and Hobert show accept/reject sampling with an exponential(9/10) candidate guarantees a finite C, which they calculate numerically. We ran their accept/reject sampler for this data, calculating C both empirically and numerically. We found both the posterior means and the Monte Carlo standard errors for the λ_i were identical, regardless of how C was estimated. Both of these methods had acceptance rates of roughly 17% . In light of our previous examples it would make sense to try a Laplace/t candidate. Notice that use of a t candidate requires automatically rejecting negative simulated variables as $\beta > 0$. Further notice that the polynomial decay of the tails of the t distribution dominates the geometric decay of the tails of $f(\beta|\mathbf{y})$, hence $C < \infty$. ESUP accept/reject sampling with this approach yielded an acceptance rate of roughly 63%. The results are summarized in Table 5.4 Again, the estimates of the posterior means for λ_i and their standard errors, after 5000 accepted values, were identical to those of KSUP rejection sampling.

In our next example, we consider a model of quasi-symmetry for the matched pairs data in Table 4.3. Recall, the sufficient statistics for the quasi-symmetry model are the margins, sum of symmetrically opposite cells and the main diagonal. We used the rounded normal instrumental density constructed in Section 4.2 as a candidate distribution for ESUP accept/reject sampling. Using the approximately

Table 5.4: Posterior means and Monte Carlo standard errors for ESUP and KSUP accept/reject sampling.

Method	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
KSUP	.071	.153	.104	.124	.627	.614
(se)	(.000)	(.001)	(.001)	(.000)	(.004)	(.002)
ESUP1	.071	.152	.103	.123	.630	.618
(se)	(.000)	(.001)	(.001)	(.000)	(.004)	(.002)
ESUP2	.070	.158	.105	.123	.630	.615
(se)	(.000)	(.001)	(.001)	(.000)	(.004)	(.002)
Method	λ_7	λ_8	λ_9	λ_{10}	β	
KSUP	.815	.836	1.29	1.83	2.48	
(se)	(.007)	(.008)	(.008)	(.005)	(.010)	
ESUP1	.820	.828	1.30	1.85	2.47	
(se)	(.007)	(.007)	(.008)	(.005)	(.010)	
ESUP2	.827	.819	1.29	1.85	2.46	
(se)	(.008)	(.007)	(.008)	(.005)	(.010)	

KSUP - KSUP accept/reject sampling with an exponential(9/10) candidate
ESUP1 - ESUP accept/reject sampling with an exponential(9/10) candidate
ESUP2 - ESUP accept/reject sampling with the Laplace/t candidate

i.i.d. variates from the ESUP algorithm, we performed a Monte Carlo exact conditional likelihood ratio test of quasi-symmetry versus the saturated model. Note that exact calculation of C for this problem would be impossible due to the complexity of the target distribution and the instrumental distribution. The result was an acceptance rate of roughly 62% with a P-value estimate of .397 (.002) after generating 100,000 instrumental variates. This value agrees closely with the values published by Booth and Butler, .393, and the results of Algorithm 4.4 (.390).

In our final example we consider the data analyzed in Mehta and Patel (1995). The data is reproduced in Table 5.5. Here the B-C index is used as a predictor for the presence or absence of Schizophrenia for 21 sets of siblings. Two sets of siblings with the same "family" predictor share the same grandparents, great-grandparents, etcetera. If $\mathbf{Y}_{ij}$ is the number of siblings with schizophrenia in family i ($i = 1, \ldots, 7$), then a possible model for this data specifies $\mathbf{Y}_{ij} \sim \text{Binomial}(n_{ij}, \pi_{ij})$

Table 5.5: Proportion of siblings with schizophrenia classified by family and B-C index.

Family	B-C Index	Prop With Schizo	Family	B-C Index	Prop With Schizo
1	15	1/1	3	1	0/1
1	7	0/1	4	2	1/1
1	6	0/1	4	0	0/4
1	5	0/1	5	6	0/1
1	3	0/2	5	3	1/1
1	2	0/3	5	0	1/1
1	0	0/1	6	3	0/1
2	2	1/1	6	0	1/4
2	0	0/1	7	6	1/1
3	9	1/1	7	2	0/1
3	2	0/1			

Source: Mehta and Patel (1995)

where π_{ij} satisfies

$$\log \frac{\pi_{ij}}{1-\pi_{ij}} = \beta_i + \lambda x_{ij} \tag{5.4}$$

for B-C index x_{ij}. It is easy to show that the sufficient statistic for the "family effect" β_i is $\sum_j Y_{ij} = Y_{i+}$ and the sufficient statistic for λ is $S_\lambda \equiv \sum_{ij} x_{ij} Y_{ij}$. For inference about λ, conditioning on the Y_{i+} is useful both to eliminate the β_i and to induce a positive correlation for success counts within the same family. In Chapter 6 we will see an alternative approach that models the β_i as normally distributed random effects. Conditional inference for λ replaces the traditional likelihood with the conditional likelihood

$$\begin{aligned} & P(S_\lambda = s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda) \\ = & \sum_{\{\mathbf{y} | \sum_{ij} y_{ij} x_{ij} = s_\lambda\}} \prod_{ij} \{y_{ij}!(n_{ij} - y_{ij})!\}^{-1} \exp(s_\lambda \lambda) a \\ = & \sum_{\{\mathbf{y} | \sum_{ij} y_{ij} x_{ij} = s_\lambda\}} \tilde{f}(\mathbf{y} | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda) a, \end{aligned}$$

where the constant of proportionality, a, is the inverse of the sum of $\tilde{f}(\mathbf{y}|Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda)$ over all possible $\mathbf{y}$ satisfying the sum constraints.

A candidate to simulate from the mass function proportional to $\tilde{f}(\mathbf{y}|Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda)$ can be constructed using the normal approximation of Section 4.2 as follows. Let $Y_{ij1} = Y_{ij}$ and $Y_{ij2} = n_{ij} - Y_{ij}$. A model that specifies the Y_{ijk} as independent Poisson random variables with means μ_{ijk} satisfying

$$\log \mu_{ij1} = \beta_{ij}^P + \beta_i + \lambda x_{ij} \text{ and } \log \mu_{ij2} = \beta_{ij}^P$$

will produce the same conditional distribution as (5.4) after conditioning on the sufficient statistics for the β_{ij}^P and β_i. Therefore we can use the rounded normal approximation of Section 4.2 as a candidate for ESUP accept/reject sampling. We note that this example is for illustration only, as exact draws from the null distribution are possible.

Numerical and graphical estimates of λ are obtained by maximizing and plotting the likelihood (or log-likelihood) $P(S_\lambda = s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda)$. As this likelihood can be intractable, Monte Carlo analysis instead maximizes and plots the *Monte Carlo* likelihood

$$\hat{P}(S_\lambda = s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda) = \sum_{l=1}^{N} I\left(\sum_{ij} y_{ijl} x_{ij} = s_\lambda\right) / N$$

where the y_{ijl} are simulated variates from $\tilde{f}(\mathbf{y}|Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda)$. As we require this approximation for several values of λ (for both numerical maximization and plotting) it is perhaps more efficient to generate the y_{il} for a fixed value of λ, say $\lambda = 0$, and use importance sampling to estimate the conditional likelihood for other values of λ. If $\lambda = 0$ is used to generate the y_{ijl}, then the importance sampling estimate of $P(S_\lambda = s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda)$ is

$$\hat{P}(S_\lambda = s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda) = \frac{\sum_{l=1}^{N} I(\sum_{ij} y_{ijl} x_{ij} = s_\lambda) \exp(s_\lambda \lambda)}{\sum_{l=1}^{N} \exp(\sum_{ij} y_{ijl} x_{ij} \lambda)}. \quad (5.5)$$

For the data in Table 5.5, Equation 5.5 with a Monte Carlo sample size of $N = 10,000$ using the rounded normal candidate in an ESUP accept/reject algorithm

lead to a conditional ML estimate $\hat{\lambda} = .34$ (the exact value is .33 (Mehta and Patel; 1995). Further we used (5.5) to plot the Monte Carlo conditional likelihood with error bounds in Figure 5.3 for a Monte Carlo sample size of N=1,000.

It would also be of interest to obtain an exact confidence interval for λ. This can be accomplished by obtaining upper and lower confidence limits via inverting the tests of $H_0 : \lambda = \lambda_0$ versus $H_1 : \lambda > \lambda_0$ and $H_2 : \lambda < \lambda_0$. If s_λ is the observed sufficient statistic for λ, then a conditional p-value for alternative H_1 is given by $P(S_\lambda \geq s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda_0)$ while a conditional p-value for alternative H_2 is given by $P(S_\lambda \leq s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda_0)$. We can use importance sampling, as in (5.5), to estimate these probabilities. For example, the estimate for the p-value corresponding to alternative H_1 is

$$\hat{P}(S_\lambda \geq s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda_0) = \frac{\sum_{l=1}^{N} I(\sum_{ij} y_{ijl} x_{ij} \geq s_\lambda) \exp(\sum_{ij} y_{ijl} x_{ij} \lambda)}{\sum_{l=1}^{N} \exp(\sum_{ij} y_{ijl} x_{ij} \lambda)},$$

while the corresponding estimate for the p-value for alternative H_2 is obtained by simply switching the direction of the inequality in the indicator function. Plots of the estimated p-values as a function of λ_0 are given in Figure 5.3. The values of λ_0 where the p-value functions are above .025 represents 97.5% upper and lower confidence intervals for λ. The intersection of these intervals is an exact 95% interval for λ (Hirji et al.; 1989). Obtaining the upper and lower limits exactly requires solving the equations

$$\hat{P}(S_\lambda \geq s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda_0) - .025 = 0$$

and

$$\hat{P}(S_\lambda \leq s_\lambda | Y_{i+} = y_{i+}, i = 1, \ldots, 7; \lambda_0) - .025 = 0,$$

which can be accomplished using Newton's algorithm. This results in an interval of approximately (.022,.774), compared to the exact interval (.022,.741) (Mehta and Patel; 1995).

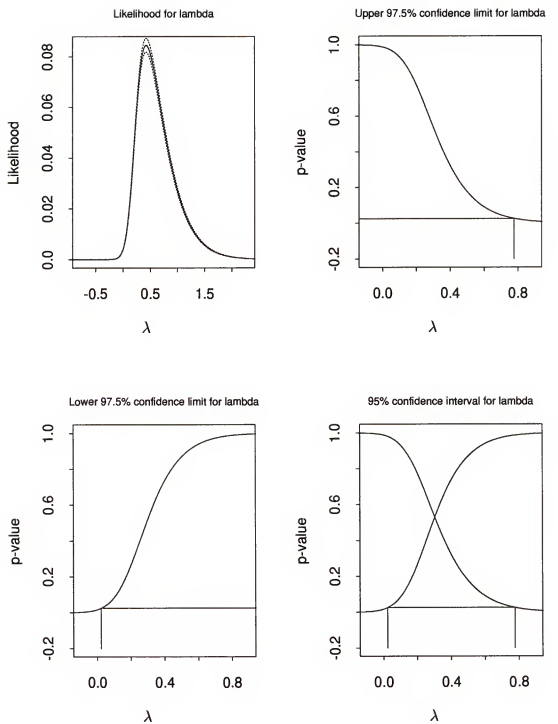

Figure 5.3: Conditional likelihood and one sided p-values for schizophrenia data.

5.7 Discussion

The arguments of this chapter show that accept/reject sampling can be fully automated, up to the choice of the candidate. That is, once a candidate is chosen, it is possible to diagnose whether $C = \infty$ and if not, C need not be calculated to perform accept/reject sampling. Also, the ESUP algorithm can be used in place of the KSUP algorithm in any situation where accept/reject sampling is now used.

In closing, we relate the ESUP algorithm to the similar accept/reject sampler of Tierney (see Section 2.6). Tierney notes that if Algorithm 2.1 is run with C replaced by a lower bound, C_{LB}, the accepted values follow a density $\tilde{f} \propto \min(f, C_{LB}g)$. He suggests using $\tilde{f}$ as the candidate distribution for an independent Metropolis sampler. An interesting problem for future research is to synthesize this method and the ESUP algorithm by adaptively updating C_{LB} in Tierney's algorithm for each accepted value. It is our conjecture that the rapid rate of convergence of C_{LB} would result in a Metropolis algorithm that always accepted and, essentially produces i.i.d. samples.

CHAPTER 6
THE MCEM ALGORITHM

6.1 Introduction

In this Chapter we discuss applications and adaptations of the Monte Carlo EM algorithm (MCEM). One focus is to show how the ESUP accept/reject sampler from the previous chapter may be applied to the EM algorithm for fitting generalized linear mixed models (referred to as GLMMs) using the Laplace approximation (Section 2.2). Further, we suggest that the Laplace approximation may itself be easily approximated extremely accurately without relying on any numerical maximization routine. This approximation comes at effectively no additional computational cost. For clustered data we suggest another adaptation of MCEM that allows the Monte Carlo sample sizes to vary depending on which clusters are more influential on the integral being estimated.

An outline of the remainder of this chapter is as follows. We begin in the next section with a brief introduction of the EM and MCEM algorithms. In the subsequent section we present unequal allocation rules for MCEM algorithms, relying heavily on classical sampling theorems and the approximations presented in Booth and Hobert (1999). In Section 6.4 we apply unequal allocation to an example and investigate its performance in a simulation study. In Section 6.5 we briefly introduce GLMMs for binary data and show how the MCEM algorithm may be used to fit them. We further show how the empirical Laplace approximation may play a role. In Section 6.6 we summarize the techniques with two examples of common GLMMs for correlated binary response data.

6.2 EM and MCEM

Suppose that a random vector $\mathbf{V}$ has joint density $f(\mathbf{v};\boldsymbol{\theta})$ depending upon a parameter vector $\boldsymbol{\theta}$. Let $\mathbf{V} = (\mathbf{Y}, \mathbf{U})$ be a partition and suppose that only the component $\mathbf{Y}$ is observable. For example, the elements of $\mathbf{U}$ may be missing observations or latent/random effects. The ML estimate of $\boldsymbol{\theta}$ based on the observable data $\mathbf{y}$ is the maximizer of the marginal density

$$\int f(\mathbf{y}, \mathbf{u}; \boldsymbol{\theta}) d\mathbf{u} \,. \tag{6.1}$$

The EM algorithm (Dempster et al.; 1977; Robert and Casella; 1999) is a widely used iterative procedure for finding the ML estimate in situations in which the integral in (6.1) is intractable, making direct maximization of the likelihood function infeasible. Let $\boldsymbol{\theta}_t$ denote the value of $\boldsymbol{\theta}$ after t iterations. Then the next value, $\boldsymbol{\theta}_{t+1}$, is obtained by maximizing the Q-function,

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = E\left\{\log f(\mathbf{y}, \mathbf{U}; \boldsymbol{\theta}) \,|\mathbf{Y} = \mathbf{y}; \boldsymbol{\theta}_t\right\} ; \tag{6.2}$$

that is, the conditional expectation of the complete data log-likelihood at $\boldsymbol{\theta}_t$. Calculating the expectation in (6.2) is referred to as the E-step. Maximizing Q with respect to $\boldsymbol{\theta}$ constitutes an M-step. If the E-step is intractable but it is possible to simulate an i.i.d. sample, $\mathbf{U}_1, \ldots, \mathbf{U}_m$, from the conditional distribution of $\mathbf{U}$ given $\mathbf{Y}$, then a Monte Carlo approximation to Q is given by

$$\tilde{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = \frac{1}{m} \sum_{j=1}^{m} \log f(\mathbf{y}, \mathbf{U}_j; \boldsymbol{\theta}) \,. \tag{6.3}$$

Notice that the right side of (6.3) is implicitly a function of $\boldsymbol{\theta}_t$ because the Monte Carlo sample is generated assuming this parameter value. In its most basic form, the Monte Carlo EM algorithm (MCEM) consists of the EM algorithm with Q replaced by $\tilde{Q}$ (see Wei and Tanner; 1990).

In this chapter we consider cases in which $(\mathbf{Y}, \mathbf{U})$ is made up of n independent components $(\mathbf{Y}_i, U_i)$, $i = 1, \ldots, n$. This situation is common in practice. For example, U_i might be an unobservable random effect associated the ith subject, domain or stratum. In such settings the Monte Carlo approximation of Q is a sum of Monte Carlo averages

$$\tilde{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} \log f(\mathbf{y}_i, U_{ij}; \boldsymbol{\theta}) = \sum_{i=1}^{n} \tilde{Q}_i(\boldsymbol{\theta}; \boldsymbol{\theta}_t) \tag{6.4}$$

where $U_{i1}, \ldots, U_{im_i}$, is an i.i.d. sample from the conditional distribution of U_i given $\mathbf{Y}_i$, simulated independently for each $i = 1, \ldots, n$. This estimate should be amended in settings where $\log f(\mathbf{y}_i, u; \boldsymbol{\theta})$ is identical (as a function of u and $\boldsymbol{\theta}$) for observations with the same values of $\mathbf{y}_i$ and explanatory variables[1] . Rather than sampling separately from each of these identical clusters, it is preferable to sample from only one and replace (6.4) with

$$\tilde{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = \sum_{i=1}^{n} \frac{c_i}{m_i} \sum_{j=1}^{m_i} \log f(\mathbf{y}_i, U_{ij}; \boldsymbol{\theta}) = \sum_{i=1}^{n} c_i \tilde{Q}_i(\boldsymbol{\theta}; \boldsymbol{\theta}_t), \tag{6.5}$$

where n now represents the number of unique clusters and c_i represents the count of times each cluster occurs in the data. Notice that it is not necessary for the sample sizes, $m_1, \ldots, m_n$, to be equal. In fact, if the conditional variances or counts of log-likelihood components differ among the n domains, which is typically the case, equal allocation of Monte Carlo resources is suboptimal. Despite this obvious fact, all applications of MCEM we are aware of utilize equal sample sizes.

In the next section we discuss a sample size allocation method which can substantially improve the performance of Monte Carlo EM algorithms. The method

[1] This occurs frequently for discrete data.

we propose can be implemented easily with most, if not all, variations of MCEM at virtually no additional computational cost.

6.3 Unequal Allocation Rules for MCEM Algorithms

The Monte Carlo approximation of Q given in (6.4) can be viewed as a stratified sample mean with equal stratum weights (Cochran; 1977, Chapter 5). Suppose that the total sample size, $M = \sum_i m_i$, is fixed and that the cost of sampling is the same in each stratum. Let

$$v_i(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t) = \mathrm{Var}\left\{\log f(\mathbf{y}_i, u_i; \boldsymbol{\theta}_{t+1}) \,|\, \mathbf{y}_i; \boldsymbol{\theta}_t\right\}. \tag{6.6}$$

Note then the variance of $\tilde{Q}$ is given by

$$\mathrm{Var}\left(\tilde{Q}\right) = \sum_{i=1}^{n} \frac{c_i^2 v_i(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)}{m_i}. \tag{6.7}$$

Then, the allocation rule that minimizes the variance of $\tilde{Q}$ is

$$m_i = M \frac{c_i \sqrt{v_i}}{\sum_k c_k \sqrt{v_k}}, \text{ for } i = 1, \ldots, n \tag{6.8}$$

(see Cochran; 1977, page 98, equation 5.23). Notice that we have suppressed the dependence of v_i on the pair, $(\boldsymbol{\theta}_{t+1}, \boldsymbol{\theta}_t)$, in (6.8). However, as the algorithm progresses $\boldsymbol{\theta}_t$ eventually stabilizes close to the ML estimate $\hat{\boldsymbol{\theta}}$. Thus, in practice one may set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t$ for the purpose of estimating the Monte Carlo error at the next iteration.

Rather than attempting to minimize the Monte Carlo error of $\tilde{Q}$ as an estimate of Q, it is perhaps more relevant to target the Monte Carlo error in the corresponding estimate of $\boldsymbol{\theta}_{t+1}$. Specifically, if $\boldsymbol{\theta}_t$ is the current value, then $\boldsymbol{\theta}_{t+1}$ is the solution of the *true* EM equation,

$$\frac{\partial}{\partial \boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = 0\,.$$

In contrast, if Q is replaced by the Monte Carlo estimate $\tilde{Q}$, then the $(t+1)$st parameter estimate, $\tilde{\boldsymbol{\theta}}_{t+1}$, is the solution of the *Monte Carlo* EM equation

$$\frac{\partial}{\partial \boldsymbol{\theta}} \tilde{Q}(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = 0\,. \tag{6.9}$$

We use Taylor series arguments similar to those in Booth and Hobert (1999), to construct a sandwich estimate of the variance of $\tilde{\boldsymbol{\theta}}_{t+1}$. Let $\tilde{Q}'(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = \frac{\partial}{\partial \boldsymbol{\theta}} \tilde{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$ and $\tilde{Q}''(\boldsymbol{\theta}; \boldsymbol{\theta}_t) = \frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \tilde{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}_t)$. Then note that

$$\tilde{Q}'(\boldsymbol{\theta}; \boldsymbol{\theta}_t) \approx \tilde{Q}'(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t) + \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)(\boldsymbol{\theta} - \boldsymbol{\theta}_{t+1}),$$

implying (as $\tilde{Q}'(\tilde{\boldsymbol{\theta}}_{t+1}; \boldsymbol{\theta}_t) = 0$)

$$\boldsymbol{\theta}_{t+1} - \tilde{\boldsymbol{\theta}}_{t+1} \approx \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)^{-1} \tilde{Q}'(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t). \tag{6.10}$$

These approximations improve when $\tilde{\boldsymbol{\theta}}_{t+1}$ is close to $\boldsymbol{\theta}_{t+1}$; that is, when there is a large Monte Carlo sample size. Equation (6.10) suggests the "sandwich" variance estimate

$$\begin{aligned} \mathrm{Var}(\tilde{\boldsymbol{\theta}}_{t+1}) &\approx \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)^{-1} \mathrm{Var}\left\{ \tilde{Q}'(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t) \right\} \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)^{-1} \\ &= \sum_i \frac{c_i^2}{m_i} \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)^{-1} \mathbf{W}_i \tilde{Q}''(\boldsymbol{\theta}_{t+1}; \boldsymbol{\theta}_t)^{-1} \\ &= \sum_i \frac{c_i^2}{m_i} \mathbf{V}_i, \end{aligned} \tag{6.11}$$

where

$$W_i = \mathrm{Var}\left\{ \frac{\partial}{\partial \boldsymbol{\theta}} \log f(\mathbf{y}_i, U_i; \boldsymbol{\theta}) | \mathbf{y}_i; \boldsymbol{\theta}_t \right\}.$$

The variance matrices, $\mathbf{V}_i$, $i = 1, \ldots, n$, can be estimated empirically using the simulated values from the previous iteration, or a smoother estimate can be obtained by averaging over several iterations. However, since the $\mathbf{V}_i$'s are matrices and not scalars the optimal allocation formula (6.8) cannot be applied directly. In

light of (6.11), good scalar summaries of $\mathbf{V}_i$ will have meaningful interpretations when summed over i. For example, setting $v_i = \text{Trace}(\mathbf{V}_i)$ in (6.8) minimizes the trace (or total variability) of the covariance matrix (6.11). If we allow $\text{ME}(\mathbf{V})$ to be the maximum eigenvalue of $\mathbf{V}$, then the relation

$$\text{ME}(\text{Var}(\tilde{\boldsymbol{\theta}}_{i+1})) \leq \sum_i \frac{c_i^2}{m_i} \text{ME}(\mathbf{V}_i)$$

suggests setting v_i equal to the maximum eigenvalue of $\mathbf{V}_i$ minimizes a bound on the maximum eigenvalue of (6.11). In contrast, setting $v_i = |\mathbf{V}_i|$ does not minimize anything meaningful as the scaled sum of determinants ($\sum_i \frac{c_i^2}{m_i} |\mathbf{V}_i|$) does not have a relevant interpretation. In Section 6.4 we apply the optimal allocation formula (6.8) setting v_i to both the trace and the maximum eigenvalue of $\mathbf{V}_i$. It is important to note that (6.11) is not invariant to re-parameterization. If $\boldsymbol{\theta}$ is not on a scale that has a meaningful interpretation, it is perhaps preferable to use (6.6), as that variance summary is invariant to any re-parameterization.

6.4 ML Estimation for the Beta-Binomial Model

The data in Table 6.1 concerns pregnancy rates for women under 18 in 13 North Central Florida counties over the three year period 1989-1991 (Gainesville Sun, April 30, 1994). The empirical variability in the child pregnancy rates among the counties is far greater than binomial sampling with a common underlying rate (deviance = 89.86 with df = 12). As an alternative one might consider a beta-binomial model in which the child pregnancy rates vary randomly among the counties according to a beta distribution; that is, the pairs (y_i, u_i), $i = 1, \ldots, 13$, are independent with distributions determined by the hierarchical model:

$$u_i \sim \text{Beta}(\alpha, \beta), \quad y_i | u_i \sim \text{Binomial}(n_i, u_i).$$

This model implies the y_i's are independent beta-binomial variables whose distribution is known explicitly (see e.g. McCullagh and Nelder; 1989, problem 4.17).

Table 6.1: Pregnancy rates for women under 18 in 13 North Central Florida counties over the three year period 1989-1991.

y_i	275	50	110	104	21	8	4	7	30	243	129	38	22
n_i	8544	1032	4851	2064	480	399	513	198	1050	8259	2946	1053	405

y_i =number of births to women under 18, n_i =total number of births Source: Gainesville Sun, April 30, 1994

The ML estimate, $(\hat{\alpha}, \hat{\beta}) = (9.947, 240.8)$, can be found by directly maximizing the likelihood so that the EM algorithm is not required. However, for illustration purposes, we consider implementation of MCEM treating the county specific rates, the u_i's, as missing data.

The log complete-data likelihood for the beta-binomial model is a sum of components of the form (up to additive constants):

$$\begin{aligned} &\log f(y_i, u_i; \alpha, \beta) \\ &= -\log B(\alpha, \beta) + (y_i + \alpha - 1)\log(u_i) + (n_i - y_i + \beta - 1)\log(1 - u_i)\,, \end{aligned} \tag{6.12}$$

where $B(\alpha, \beta)$ is the complete beta function. The conditional distribution of u_i given y_i is beta with parameters $y_i + \alpha$ and $n_i - y_i + \beta$. Differentiation of (6.12) yields

$$\tilde{Q}_{\theta\theta} = -\begin{bmatrix} \psi'(\alpha+\beta) - \psi'(\alpha) & \psi'(\alpha+\beta) \\ \psi'(\alpha+\beta) & \psi'(\alpha+\beta) - \psi'(\beta) \end{bmatrix}. \tag{6.13}$$

where ψ denotes the digamma function. In this example, further calculations also reveal the center of the sandwich variance estimate,

$$\mathbf{W}_i = \begin{bmatrix} \psi'(\alpha_t + \beta_t + n_i) - \psi'(\alpha_t + y_i) & \psi'(\alpha_t + \beta_t + n_i) \\ \psi'(\alpha_t + \beta_t + n_i) & \psi'(\alpha_t + \beta_t + n_i) - \psi'(\beta_t + n_i - y_i) \end{bmatrix} \tag{6.14}$$

Plugging the ML values into the Monte Carlo variance of the parameter estimates (6.11) yields the theoretical gain in efficiency over equal allocation when the estimates are close to convergence. For this example, using either the trace or

maximum eigenvalue criterion, the estimated Monte Carlo variance for unequal allocation is 66% that of equal allocation.

To assess the performance of unequal allocation rules in practice we performed multiple runs of the algorithm for a fixed number of EM steps using the pregnancy data and beta binomial model from Section 6.3. Specifically, we ran 1,000 independent runs of 100 MCEM iterations, with a Monte Carlo sample size of $M = 1,000$ at each iteration. The number of EM iterations (100) was chosen to well exceed what was required for the parameter estimates to stabilize for this example. On the other hand, the number of independent runs (1,000) was chosen to accurately estimate the variance-covariance matrix for the parameter estimates in repeated applications of EM. Specifically, 1000 runs is a lower bound on the number required to estimate the 2×2 variance-covariance with a 10% margin-of-error see (see Booth and Sarkar; 1998, section 5).

To mimic practical situations where MCEM would be required, the variance matrix W_i was estimated with the generated sample from the previous iteration despite an available closed form expression (6.14). The computational cost of this estimation was minimal and simply involved saving the relevant sums during the generation of the beta-variates. This approach results in an estimate of V_i for the previous iteration, which is sufficient for the purpose of allocation. Smoothed estimates of V_i obtained by averaging information from previous iterations were explored, but did not provide any improvement.

Table 6.2 gives various summaries of performance for the MCEM algorithm for each of the allocation schemes. Notice that unequal allocation using either the maximum eigenvalue or the trace of V_i improves on equal allocation in every measure that was considered. As predicted the trace and maximum eigenvalue of the Monte Carlo variance matrix (6.11) are around 66% the values obtained by equal allocation. It was particularly encouraging that the improvement remained

Table 6.2: Variance summaries and mean squared error of parameter estimates based on 1000 independent runs of 100 MCEM steps, with a Monte Carlo sample size of 1,000 within each MCEM step.

	Allocation Rule		
Performance Summary	Equal	Trace of V_i	Max eigen of V_i
Var($\hat{\alpha}$)	.090	.065	.063
Var($\hat{\beta}$)	57.7	40.8	39.2
Trace of Var($\hat{\alpha}$ $\hat{\beta}$)$'$	57.8	40.8	39.3
Max Eigen of Var($\hat{\alpha}$ $\hat{\beta}$)$'$	57.8	40.8	39.3
Determinant of Var($\hat{\alpha}$ $\hat{\beta}$)$'$	.129	.104	.103
MSE for $\hat{\alpha}$	.090	.065	.063
MSE for $\hat{\beta}$	57.8	41.2	39.2

constant regardless of which performance measure the allocation scheme targeted. To be fair, both the trace and the maximum eigenvalue are dominated by the Monte Carlo variability in $\hat{\beta}$ in this example. However, similar results were obtained when the binomial sample sizes, n_i, were dropped by a factor of ten so that $\hat{\alpha}$ and $\hat{\beta}$ had a relatively equal contribution to the overall Monte Carlo variability.

6.5 MCEM for Logit/Normal GLMMs

In this section we discuss a model where the "E" step of the EM algorithm is intractable and suggest the use of ESUP rejection sampling with optimal allocation as an alternative to numerical integration. In particular, we consider a generalized linear mixed model (GLMM) for correlated binary response data. A general review of GLMMs for categorical response data is given in Agresti et al. (2000). GLMMs are specified using a two stage hierarchy. For our applications, the top level of the hierarchy assumes binary responses, $Y_{ij}|U_i$, are mutually independent Bernoulli random variables for $i = 1, \ldots, I$ and $j = 1, \ldots, n_i$ with success probabilities, π_{ij}, satisfying

$$\eta_{ij} = \log \frac{\pi_{ij}}{1 - \pi_{ij}} = \mathbf{X}_{ij}^t \boldsymbol{\beta} + U_i, \tag{6.15}$$

for fixed effects $\boldsymbol{\beta}$, vector $\mathbf{X}_{ij}$, and random effect U_i. To emphasize the dependence on $\boldsymbol{\beta}$ and U_i we will often write π_{ij} as $\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta} + U_i)$. The U_i are assumed to be mutually independent $\mathrm{N}(0, \sigma^2)$ random variables. More general GLMMs for clustered data replace (6.15) with

$$\eta_{ij} = \log \frac{\pi_{ij}}{1 - \pi_{ij}} = \mathbf{X}_{ij}^t\boldsymbol{\beta} + \mathbf{Z}_{ij}^t\mathbf{U}_i, \tag{6.16}$$

where $\mathbf{U}_i \sim \mathrm{N}(0, \boldsymbol{\Sigma})$ for vectors $\mathbf{Z}_{ij}$ and positive definite matrix $\boldsymbol{\Sigma}$ of variance components. Quite often it is the case that $\mathbf{Z}_{ij} = \mathbf{Z}_i$, that is the random effect design matrix does not depend on j. This model can be fit exactly as model (6.15) with σ^2 replaced by $\mathbf{Z}_i^t\boldsymbol{\Sigma}\mathbf{Z}_i$ in all calculations. A further generalization specifies that the complete vector of Bernoulli log-odds success probabilities satisfies

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{U},$$

with $\mathbf{U} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$. However, as our examples focus on clustered data, we confine our attention to (6.15) and (6.16).

Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{in_i})$ and $Y_{i+} = \sum_j^{n_i} Y_{ij}$. The ML estimates for $\boldsymbol{\beta}$ and σ are the maximizers of the marginal log-likelihood,

$$\sum_{i=1}^{I} \log \int_{\mathbb{R}} (2\pi\sigma^2)^{-1/2} \exp\left[\sum_{j=1}^{n_i} y_{ij}\mathbf{X}_{ij}^t\boldsymbol{\beta} + y_{i+}u_i - \frac{u_i^2}{2\sigma^2}\right] \prod_{j=1}^{n_i} \left(1 + \exp\left[\mathbf{X}_{ij}^t\boldsymbol{\beta} + u_i\right]\right)^{-1} du_i,$$

or for (6.16)

$$\sum_{i=1}^{I} \log \int_{\mathbb{R}^k} (2\pi)^{-k/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left[\sum_{j=1}^{n_i} y_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta} + \mathbf{Z}_{ij}^t\mathbf{u}_i) - \mathbf{u}_i^t\boldsymbol{\Sigma}^{-1}\mathbf{u}_i/2\right] \prod_{j=1}^{n_i} \left(1 + \exp\left[\mathbf{X}_{ij}^t\boldsymbol{\beta} + \mathbf{Z}_{ij}^t\mathbf{u}_i\right]\right)^{-1} d\mathbf{u}_i.$$

These I integrals are intractable, and thus require the use of numerical or Monte Carlo integration. Numerical integration via Gauss Hermite quadrature is the method used by the SAS procedure NLMIXED. McCulloch (1997) reviews Monte

Carlo fitting algorithms for GLMMs while Booth and Hobert (1999) focus specifically on the MCEM algorithm.

The EM algorithm for this model treats the U_i as missing data. If $\boldsymbol{\beta}_t$ and σ_t ($\boldsymbol{\Sigma}_t$) are the current values of the algorithm, the next estimates, $\boldsymbol{\beta}_{t+1}$ and σ_{t+1} ($\boldsymbol{\Sigma}_{t+1}$), are the maximizers of the Q function (see Section 6.2):

$$Q(\boldsymbol{\beta}, \sigma; \boldsymbol{\beta}_t, \sigma_t) = -I \log \sigma - \sum_{i=1}^{I} \frac{E_t^*[U_i^2]}{2\sigma^2} + \tag{6.17}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{n_i} y_{ij} \mathbf{X}_{ij}^t \boldsymbol{\beta} - \sum_{i=1}^{I} \sum_{j=1}^{n_i} E_t^* \left[\log \left(1 + \exp \left[\mathbf{X}_{ij}^t \boldsymbol{\beta} + U_i \right] \right) \right] \tag{6.18}$$

or for (6.16) we have

$$Q(\boldsymbol{\beta}, \boldsymbol{\Sigma}; \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_t) = - \frac{I}{2} \log |\boldsymbol{\Sigma}| - \sum_{i=1}^{I} E_t^*[\mathbf{U}_i^t \boldsymbol{\Sigma} \mathbf{U}_i]/2$$

$$+ \sum_{i=1}^{I} \sum_{j=1}^{n_i} y_{ij} \mathbf{X}_{ij}^t \boldsymbol{\beta} - \sum_{i=1}^{I} \sum_{j=1}^{n_i} E_t^* \left[\log \left(1 + \exp \left[\mathbf{X}_{ij}^t \boldsymbol{\beta} + \mathbf{Z}_{ij}^t \mathbf{U}_i \right] \right) \right],$$

where E_t^* denotes expectation with respect to the distribution of $U_i | \mathbf{Y}_i = \mathbf{y}_i; \boldsymbol{\beta}_t, \sigma_t$ or $\mathbf{U}_i | \mathbf{Y}_i = \mathbf{y}_i; \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_t$ (respectively). These densities are given by

$$\begin{aligned} f(u_i | \mathbf{Y}_i = \mathbf{y}_i; \boldsymbol{\beta}_t, \sigma_t) &\propto f(\mathbf{y}_i, u_i; \boldsymbol{\beta}_t, \sigma_t) \\ &\propto \exp \left[y_{i+} u_i - \frac{u_i^2}{2\sigma_t^2} \right] \prod_{j=1}^{n_i} \left(1 + \exp \left[\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + u_i \right] \right)^{-1} \end{aligned} \tag{6.19}$$

$$\begin{aligned} f(\mathbf{u}_i | \mathbf{Y}_i = \mathbf{y}_i; \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_t) &\propto f(\mathbf{y}_i, \mathbf{u}_i; \boldsymbol{\beta}_t, \boldsymbol{\Sigma}_t) \\ &\propto \exp \left[\sum_{j=1}^{n_i} y_{ij} \mathbf{Z}_{ij}^t \mathbf{u}_i - \mathbf{u}_i^t \boldsymbol{\Sigma}_t \mathbf{u}_i / 2 \right] \prod_{j=1}^{n_i} \left(1 + \exp \left[\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t \mathbf{u}_i \right] \right)^{-1}. \end{aligned}$$

Notice that the Q function can be maximized independently for the variance components and the fixed effects respectively. Further, the maximizer of (6.17) for σ^2 has the closed form expression,

$$\sigma_{t+1}^2 = \sum_{i=1}^{I} E_t^*[U_i^2]/I.$$

The estimate of the variance components for model (6.16) requires maximizing

$$-\frac{I}{2}\log|\boldsymbol{\Sigma}| - \sum_{i=1}^{I} E_t^*[\mathbf{U}_i^t\boldsymbol{\Sigma}\mathbf{U}_i]/2$$

which is much more difficult. However, as discussed earlier, if $\mathbf{Z}_{ij} = \mathbf{Z}_i$ then it is much easier to obtain $\boldsymbol{\Sigma}_{t+1}$ as the maximizer of

$$-\frac{1}{2}\sum_{i=1}^{I}\log(\mathbf{Z}_i^t\boldsymbol{\Sigma}\mathbf{Z}_i) - \sum_{i=1}^{I}\frac{E_t^*[U_i^2]}{2\mathbf{Z}_i^t\boldsymbol{\Sigma}\mathbf{Z}_i},$$

where U_i is from the density (6.19) with σ replaced by $\mathbf{Z}_i^t\boldsymbol{\Sigma}\mathbf{Z}_i$. With an estimate of $E_t^*[U_i^2]$, existing algorithms for fitting linear mixed models could be used to maximize this equation.

From (6.18) we see the maximizer of Q for $\boldsymbol{\beta}$ satisfies

$$\frac{\partial Q(\boldsymbol{\beta},\sigma;\boldsymbol{\beta}_t,\sigma_t)}{\partial\boldsymbol{\beta}} = \sum_{i=1}^{I}\sum_{j=1}^{n_i}\left\{y_{ij} - E_t^*[\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)]\right\}\mathbf{X}_{ij} = \mathbf{0}.$$

which may be equivalently stated in vector notation by

$$\frac{\partial Q(\boldsymbol{\beta},\sigma;\boldsymbol{\beta}_t,\sigma_t)}{\partial\boldsymbol{\beta}} = \mathbf{X}^t\left\{\mathbf{y} - E_t^*[\boldsymbol{\pi}]\right\} = \mathbf{0}. \tag{6.20}$$

Further note that the Hessian matrix also follows a convenient form:

$$\begin{aligned}\frac{\partial^2 Q(\boldsymbol{\beta},\sigma;\boldsymbol{\beta}_t,\sigma_t)}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^t} &= \sum_{i=1}^{I}\sum_{j=1}^{n_i} E_t^*\left\{\mathbf{X}_{ij}\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)[1-\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)]\right\}\mathbf{X}_{ij}\mathbf{X}_{ij}^t \\ &= \mathbf{X}^t D_t\mathbf{X} \end{aligned} \tag{6.21}$$

where D_t is a diagonal matrix with diagonal elements $E_t^*[\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)(1 - \pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i))]$. These derivatives are equivalent for model (6.16) with $\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)$ replaced by $\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t\mathbf{U}_i)$. Therefore, to perform the EM algorithm we must know, $E_t^*[U_i^2]$, $E_t^*[\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)]$ and $E_t^*\left\{\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)[1-\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + U_i)]\right\}$ for model (6.15) and $E^*[\mathbf{U}_i^t\boldsymbol{\Sigma}_t\mathbf{U}_i]$, $E_t^*[\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t\mathbf{U}_i)]$ and $E_t^*\left\{\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t\mathbf{U}_i)[1-\pi_{ij}(\mathbf{X}_{ij}^t\boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t\mathbf{U}_i)]\right\}$ for model (6.16).

As all of these expectations are intractable we use a Monte Carlo approximation. Exact simulation from (6.19) is possible by accept/reject sampling using a $N(0, \sigma_t^2)$ or $N(0, \mathbf{\Sigma}_t^2)$ candidate. However, similar to the example in Section 5.6, use of the marginal normal distribution of the U_i as a candidate distribution for an accept/reject sampler can be inefficient, especially when the n_i are large. We instead use ESUP rejection sampling with a candidate obtained by shifting and scaling a t_3 variate by the Laplace approximation mean and standard deviation. Recall (see Section 2.2) that the Laplace approximation mean, denoted by μ_i or $\boldsymbol{\mu}_i$, is the mode of $f(u_i|\mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t)$ or $f(\mathbf{u}_i|\mathbf{y}_i, \boldsymbol{\beta}_t, \mathbf{\Sigma}_t)$ respectively. We may write, more specifically, that μ_i is the value of u_i satisfying

$$\frac{\partial}{\partial u_i} f(u_i, \mathbf{y}_i; \boldsymbol{\beta}_t, \sigma_t) = y_{i+} - u_i/\sigma_t^2 - \sum_{j=1}^{n_i} \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + u_i) = 0,$$

and $\boldsymbol{\mu}_i$ is the value of $\mathbf{u}_i$ satisfying

$$\sum_{j=1}^{n_i} y_{i+}\mathbf{Z}_{ij} - \mathbf{\Sigma}_t^{-1}\mathbf{u}_i - \sum_{j=1}^{n_i} \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t \mathbf{u}_i)\mathbf{Z}_{ij} = 0.$$

Further we may write the the Laplace variance explicitly as

$$\rho_i = \sigma_t^2 \left(1 + \sigma_t^2 \sum_{j=1}^{n_i} \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mu_i)(1 - \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mu_i))\right)^{-1},$$

and

$$\boldsymbol{\rho}_i = \left(\mathbf{\Sigma}_t^{-1} + \sum_{j=1}^{n_i} \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t \boldsymbol{\mu}_i)[1 - \pi_{ij}(\mathbf{X}_{ij}^t \boldsymbol{\beta}_t + \mathbf{Z}_{ij}^t \boldsymbol{\mu}_i)]\mathbf{Z}_{ij}\mathbf{Z}_{ij}^t\right)^{-1}$$

for (6.15) and (6.16) respectively.

ESUP rejection sampling generates from the shifted and scaled t_3 distribution (with density labeled g) and accepts those candidates U_i such that a random uniform is less than $f(\mathbf{y}_i, U_i; \boldsymbol{\beta}_t, \sigma_t)/\hat{C}_i g(U_i)$ where $\hat{C}_i$ is the largest observed value of $f(\mathbf{y}_i, U_i; \boldsymbol{\beta}_t, \sigma_t)/g(U_i)$ (with the obvious changes for model (6.16)). The exact supremum is finite because, as mentioned earlier, a normal candidate leads to a

finite supremum and the tails of the t dominate the tails of a normal (in both the univariate and multivariate case). Notice that, in the process of using the generated U_i to estimate (6.20) and (6.21), one could also obtain an empirical estimate of μ_i with the value of U_i that produces the largest $f(\mathbf{y}_i, U_i; \boldsymbol{\beta}_t, \sigma_t)$. This estimate and the corresponding approximation to the Laplace standard deviation could be used to avoid the numerical maximization routine required to calculated the μ_i exactly. Although this approximation was extremely accurate in the examples we considered, the computational gains are often small as the convergence to μ_i via Newton/Raphson is generally very rapid.

In Algorithm 6.1 we outline the algorithm for fitting GLMMs using the empirical Laplace approximation and unequal allocation for model (6.15).

6.6 Examples

We performed ESUP rejection sampling with the empirical Laplace approximation using the data simulated by Booth and Hobert (1999), reproduced in Table 6.3. Here, the data represents the success or failure of a treatment given at 15 dosages in 10 centers to 150 subjects, each patient receiving only one dosage. Booth and Hobert's model specifies

$$\log \frac{\pi_{ij}}{1-\pi_{ij}} = x_j\beta + U_i,$$

where x_j refers to dosage j. Their analysis reports exact values of $\hat{\beta} = 6.1322$ and $\hat{\sigma} = 1.3291$. which are confirmed by NLMIXED.

The MCEM sample paths, $\{\tilde{\beta}_t\}$ and $\{\tilde{\sigma}_t\}$ obtained using ESUP rejection sampling and the empirical Laplace approximation are plotted in Figure 6.1. A total Monte Carlo sample size of $M = 100,000$ (10,000 per cluster) was used within each EM iteration. The ML estimates are indicated with solid lines in the two plots.

Algorithm 6.1 MCEM Algorithm for Fitting Binary Response GLMMs

Set $\widehat{E_t^*}[U_i^2] = 0$

1 For $i = 1$ to I

For $k = 1$ to m_i

Do until the candidate is accepted

Simulate candidate $U_c = \mu_i + \sqrt{\rho_i} T_3$

Set $W_i = f(U_c, \mathbf{y}_i; \boldsymbol{\beta}_t, \sigma_t)/g(T_3)$ for g the t_3 density

If a random uniform is $\leq W_i/\hat{C}_i$ then set $U_{ik} = U_c$

Update empirical supremum $\hat{C}_i = \max\left(\hat{C}_i, W_i\right)$

end

Update $\widehat{E_t^*}[U_i^2] = \widehat{E_t^*}[U_i^2] + U_c/m_i$

Update the sums necessary for optimal allocation

end

end

Update $\sigma_{t+1} = \left(\sum_{i=1}^{I} \widehat{E_t^*}[U_i^2]\right)/I$

Comment Update $\boldsymbol{\beta}_{t+1}$

Starting value is the last value $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t$

Do until $||Hess^{-1} Deriv|| < \epsilon$ plus one extra iteration

Set $\widehat{E_t^*}\pi_{ij} = 0$ and $\widehat{E_t^*}\pi_{ij}(1 - \pi_{ij}) = 0$

For $i = 1$ to I

For $k = 1$ to m_i

For $j = 1$ to n_i

Set $\pi_{ij} = \pi_{ij}(\boldsymbol{\beta}_{t+1}, U_{ik})$

$\widehat{E_t^*}\pi_{ij} = \widehat{E_t^*}\pi_{ij} + \pi_{ij}/m_i$

$\widehat{E_t^*}\pi_{ij}(1 - \pi_{ij}) = \widehat{E_t^*}\pi_{ij}(1 - \pi_{ij}) + \pi_{ij}(1 - \pi_{ij})/m_i$

end

If at final iteration keep track of:

$\mu_i = \text{argmax}_{U_{ik}} f(U_{ik}, \mathbf{y}, \boldsymbol{\beta}_{t+1}, \sigma_{t+1})$

Sums necessary for optimal allocation

end

If at final iteration

update $\rho_i = \sigma_{t+1}^2 \left(1 + \sigma_{t+1}^2 \sum_{j=1}^{n_i} \pi_{ij}(\boldsymbol{\beta}_t, \mu_i)(1 - \pi_{ij}(\boldsymbol{\beta}_t, \mu_i))\right)^{-1}$

end

$Deriv = \mathbf{X}^t(\mathbf{y} - \widehat{E_t^*}\boldsymbol{\pi})$

$Hess = \mathbf{X}^t \hat{D}_t \mathbf{X}$

Set $\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_{t+1} - Hess^{-1} Deriv$

end

Update the m_i

Return to step 1 until the change in the $\boldsymbol{\beta}$ and σ estimates is small

Table 6.3: Booth and Hobert's simulated data.

	Dosage							
Center	1/15	2/15	3/15	4/15	5/15	6/15	7/15	8/15
1	1	0	0	0	0	1	1	0
2	0	1	1	1	1	1	1	1
3	0	1	0	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	0	1	1	1	1	1	1	1
6	0	0	0	1	0	1	1	1
7	0	1	0	0	1	1	1	1
8	1	1	1	1	1	1	1	1
9	1	0	0	1	1	0	1	1
10	1	1	1	1	1	1	1	1
	9/15	10/15	11/15	12/15	13/15	14/15	1	Sum
1	1	1	1	1	1	1	1	10
2	1	1	1	1	1	1	1	14
3	1	1	1	1	1	1	1	13
4	1	1	1	1	1	1	1	15
5	1	1	0	1	1	1	1	13
6	0	1	1	1	1	1	1	10
7	1	1	1	1	1	1	1	12
8	1	1	1	1	1	1	1	15
9	1	1	1	1	1	1	1	12
10	1	1	1	1	1	1	1	15

Source: Booth and Hobert (1999)

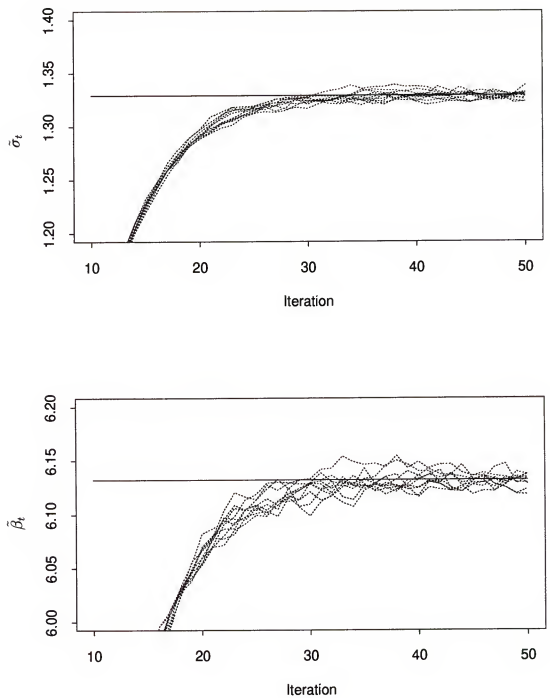

Figure 6.1: Sample path plots $\tilde{\beta}_t$ and $\tilde{\sigma}_t$ by MCEM iteration.

To check the performance of the empirical Laplace approximation, we compared it to the exact Laplace approximation (calculated numerically) for 5,000 MCEM iterations with the first iteration started at the ML values. Again we set $M = 100,000$, or a Monte Carlo sample size of 10,000 per cluster. We calculated

$$\sum_{t=1}^{5,000} |\hat{\mu}_{it} - \mu_{it}|/5,000$$

where $\hat{\mu}_{it}$ is the empirical Laplace mean for cluster i at iteration t and μ_{it} is the exact Laplace mean (calculated numerically to 10 decimal places). These average absolute deviations of the empirical Laplace approximation to the true Laplace approximation varied from .0001 to .00007. That is, the empirical Laplace approximation generally agreed with the exact up to four decimal places.

The next example concerns the data in Table 6.4 (taken from the General Social Survey, http://www.icpsr.umich.edu/GSS/) which cross-classifies 1,850 adults responses to 3 questions on abortion:

> Please tell me whether or not you think it should be possible for a pregnant woman to obtain a legal abortion if
>
> 1. If the family has a very low income and cannot afford any more children
> 2. The woman wants it for any reason
> 3. If she is not married and does not want to marry the man

by gender. Surprisingly there are no zero cell counts, despite the obvious inconsistency, for example, in answering yes to question 2 but no to questions 1 or 3. However, as one would expect, the majority of the respondents answered all yes or all no. A potential GLMM for this data models the yes/no responses as conditionally independent Bernoulli trials. Specifically, let $Y_{ij}|U_i$ be independent Bernoulli

Table 6.4: Cross-classification of 1,850 adults' responses to three questions on whether or not a women has the right to have an abortion by gender.

	Question					Question			
Gender	1	2	3	Count	Gender	1	2	3	Count
Male	yes	yes	yes	342	Female	yes	yes	yes	440
Male	yes	yes	no	26	Female	yes	yes	no	25
Male	yes	no	yes	11	Female	yes	no	yes	14
Male	yes	no	no	32	Female	yes	no	no	47
Male	no	yes	yes	6	Female	no	yes	yes	14
Male	no	yes	no	21	Female	no	yes	no	18
Male	no	no	yes	19	Female	no	no	yes	22
Male	no	no	no	356	Female	no	no	no	457

trials with success probabilities, π_{ij}, satisfying

$$\begin{aligned} \text{logit}(\pi_{ij}) &= \text{Intercept} + \text{Gender} + \text{Question} + \text{Person} \\ &= \alpha + \gamma I(\text{person } i \text{ is female}) + \beta_j + U_i \end{aligned} \tag{6.22}$$

where I represents the indicator function. The U_i are assumed to be independent $N(0, \sigma^2)$ random variables. For model identifiability we set $\beta_3 = 0$. The assumption of normality for the random effect in this model may not be the best choice. A non-parametric approach that allows the $u_i = \pm\infty$ with positive probability might be preferable due to the large counts in the all yes and all no cells. Aitkin (1999, 1996) gives details on the non-parametric approach.

This example is ideal for optimal-allocation as the counts (c_i) vary greatly. The standard deviation of the counts is 165.6 with the largest being 457 and the smallest being 6. We repeatedly ran the Algorithm 6.1 on this dataset starting at the exact ML values; that is, we set $\alpha_0 = \hat{\alpha}$, $\gamma_0 = \hat{\gamma}$, etc. Repeatedly applying the algorithm with these starting values allows us to study the "pure" Monte Carlo error for each allocation method. Table 6.5 summarizes this error by reporting the variance and mean squared errors of the parameter estimates after the MCEM algorithm has preceded two iterations past the starting values. Three choices of v_i were used in (6.8): equal allocation $v_i = 1/c_i$, allocation by the counts alone

Table 6.5: Variance (Var), mean squared errors (MSE) and ratio to equal allocation (in parentheses) of parameter estimates by allocation method for Monte Carlo sample size $M = 1,000$.

	Allocation Method					
	$v_i = 1/c_i$		$v_i = 1$		(6.6)	
Parameter	Var	MSE	Var	MSE	Var	MSE
α	.003	.003	.002 (.62)	.002 (.58)	.002 (.58)	.002 (.54)
γ	.006	.006	.003 (.55)	.003 (.55)	.003 (.49)	.003 (.49)
β_1	.000	.000	.000 (.30)	.000 (.31)	.000 (.39)	.000 (.39)
β_2	.000	.000	.000 (.30)	.000 (.31)	.000 (.38)	.000 (.39)
σ	.014	.014	.004 (.27)	.004 (.28)	.004 (.31)	.004 (.32)

$v_i = 1$ and finally using (6.6). Notice allocating by the counts or (6.6) improves over equal allocation for every summary considered. In parentheses, Table 6.5 gives the variance and mean squared error of the parameter estimates for the two non-equal allocation schemes divided by the variance and mean squared error for equal allocation (respectively). For example the variance of the α parameter estimate for allocation by the counts alone was 62% that of equal allocation. The most drastic improvement for this allocation scheme was for σ where the variance and mean squared error were 27% and 28% that of equal allocation respectively. The improvements attained using allocation by (6.6) were similar.

It is not surprising that allocation by the counts alone performed so well in this example, as the conditional distributions of u_i given (y_{i1}, y_{i2}, y_{i3}) varied little from cluster to cluster, making the counts the dominant factor in (6.8). Sample path plots of the parameter estimates by MCEM iteration were constructed (Figure 6.2). Notice the drastic improvement in estimating σ. Again, the plots suggests either of the two unequal allocation methods greatly improves on equal allocation.

6.7 Discussion

The two computing methods developed in this chapter, unequal allocation and the empirical Laplace approximation, add virtually no computing time to the MCEM algorithm. In fact, Booth and Hobert (1999) argue that the extra step of

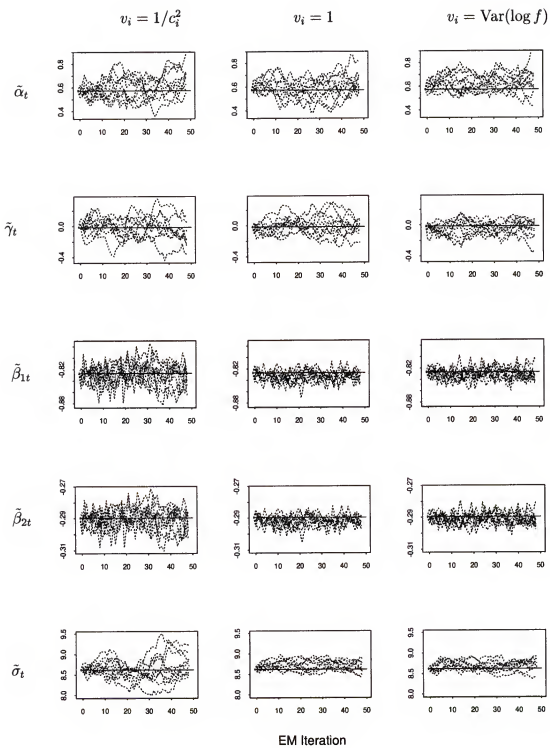

Figure 6.2: Sample path plots of parameter estimates by MCEM iteration for the abortion data.

estimating the Monte Carlo variability in the parameter estimates (required for optimal allocation) should always be performed in order to control the Monte Carlo error relative the EM step size. This control can be achieved by increasing the sample size, M, when necessary. Since the sample size allocation rules are applied separately at each iteration, the methods described here can be applied directly with no modification if M is increased as the algorithm progresses.

We will concede the point that both techniques are more useful when one is programming with a "looping mentality". Some programming languages are not as effective using loops and avoid them whenever possible with clever uses of matrix arithmetic or pre-written functions (for example Splus and R). When writing code in these languages the gains from optimal allocation and the empirical Laplace approximation may be offset by the added computing time caused by using loops. However, for problems of sufficient size and complexity, loops invariably become necessary.

CHAPTER 7
DISCUSSION

In this final chapter, we summarize the work of the dissertation and present some ideas for future research. We break the discussion into two sections, the first covering exact conditional testing and the second accept/reject sampling and the MCEM algorithm.

7.1 Exact Conditional Analysis

In the dissertation we reviewed exact conditional analysis and presented a new algorithm for generating from the null conditional distribution arising from lack of fit tests for log-linear models. Our Markov chain algorithm uses "local-updates" that slightly perturb the current state of the chain. This algorithm and Booth and Butler's (1999) algorithm become more efficient as the cell counts increase because the normal approximation becomes more accurate. In contrast, most other Monte Carlo and Markov chain Monte Carlo approaches in this area, as well as all enumeration techniques, become more difficult to implement as the cell counts increase. It could be argued, as large sample approximations become more accurate as cell counts increase, that data with large cell counts are not of interest for Monte Carlo exact tests. However, there are situations, for example when presenting statistical evidence in court cases, where the guarantee of a type I error rate is required regardless of the size of the cell counts. Therefore the algorithm in this dissertation and Booth and Butler's algorithm are as much compliments to other Monte Carlo and enumeration techniques as they are competitors.

The algorithm in this dissertation is representative of a recent direction of research in Monte Carlo exact conditional analysis, Markov chain Monte Carlo

algorithms that slightly perturb the current table (state). For tables with large dimensions, these kinds algorithms are more effective than other approaches. The two major issues with "local update algorithms" are irreducibility and convergence control. The Markov basis approach of Diaconis and Sturmfels (1998) guarantees irreducibility. However it involves computationally heavy preprocessing to obtain the Markov basis. It is possible, that the Markov bases for a class of models can be characterized, thus eliminating this preprocessing. For example, in Section 3.7 we characterize the Markov bases for all complete symmetry models for $I \times I$ contingency tables. Obtaining a more general result for a large class of models would be a difficult problem for future research. For example, one might be able to characterize the Markov bases for classes of graphical models, such as decomposable models (Lauritzen; 1996).

Convergence control for MCMC exact conditional analysis has not been developed at all. Diaconis and Sturmfels (1998) selected every 500th variate and treated them as independent. Smith et al. (1996a) and the work in this dissertation used batching. These ad-hoc methods are unsatisfactory, as we have seen that the autocorrelations in the Markov chain can vary greatly from one problem to the next. Further, neither produce a consistent estimate of the limiting variance of ergodic averages from the Markov chain. As discussed in Chapter 4, regenerative simulation, (Mykland et al.; 1995) relies on a tight minorization condition which may be difficult or impossible to establish in realistic settings. Window estimators (Geyer; 1992) rely on the choice of a window and require storing the entire chain. Establishing a practical and consistent estimate of Monte Carlo error is currently the biggest challenge for MCMC exact conditional analysis. As another possible direction, note that generating from the Markov chain for these problems is often very fast relative to the time required to evaluate a point of the chain. A good example of this is lack-of-fit tests for non-saturated alternatives. Here, the number

of operations to calculate the lack-of-fit statistic can far exceeds the number of operations to generate a point from the Markov chain. This is an ideal situation for sub-sampling the Markov chain, in which case a bound on the mixing time would be invaluable. However, Diaconis and Sturmfels (1998) suggest that the complex Markov chains used in this area, such as the one from this dissertation, may be beyond the scope of rigorous analysis of the mixing time.

Using Groebner bases to simulate from exact conditional distributions is an exciting area of future research. Two main issues remain unresolved. The first is the difficulty in knowing the exact probabilities to sequentially simulate from. The second is the computational overhead in generating the Groebner basis. As mentioned earlier, the first problem may be avoided by sequentially simulating using the approximate sequential means from Chapter 4 and an importance or Metropolis step. The second problem of the computational overhead for calculating Groebner bases is severe, especially so in contingency tables where the integer constraints forces the practitioner to work with polynomials of very high degree. For example, if a table entry, y_{ij}, is between 0 and m, the polynomial

$$\prod_{l=0}^{m}(y_{ij} - l) = 0 \tag{7.1}$$

must be accounted for (this polynomial is of degree $m + 1$). This problem in mind, the usefulness of this approach remains an open question for future research.

As mentioned in Chapter 3 it is possible to use a Metropolis algorithm, similar to the one from this dissertation, to generate from the null conditional distribution for tests involving data with continuous support. We illustrate this technique through an example from Butler et al. (1999). Assume $y_{ij} \sim \text{Gamma}(\alpha_i, \beta_{2i})$ for $i = 1 \ldots I$ and $j = 1, \ldots n_i$ where $\alpha_i = \lambda_i + \beta_1$. Then under the null hypothesis $H_0 : \lambda_1 = \ldots = \lambda_I = 0$, the sums, $y_{i+} = \sum_{j=1}^{n_i} y_{ij}$, are sufficient for β_{2i}. Note that $(y_{i1}/y_{i+}, \ldots, y_{in_i}/y_{i+})$ is a symmetric Dirichlet random variable with parameter

β_1 and hence is ancillary for β_{2i}. Thus $\prod_{j=1}^{n_i} y_{ij}/y_{i+}^{n_i}$ is also ancillary for β_{2i} and therefore so is

$$z_i = n_i \log\left(\prod_{j=1}^{n_i} y_{ij}^{1/n_i}/\bar{y}_{i+}\right),$$

which is n_i times the log-ratio of the geometric and arithmetic means of y_{ij} (as j varies). As $z_+ = \sum_{i=1}^{I} z_i$ is sufficient for β_1, interest lies in calculating probabilities from $z_i|z_+$ to test the null hypothesis. Though this distribution is intractable, Butler et al. (1999) give an extremely accurate saddlepoint approximation. As numerical integration of the saddlepoint approximation is very difficult Butler et al. further suggest using importance sampling with a normal candidate distribution. The normal approximation is based on the large sample distribution of the ML estimates of the α_i conditional on y_{i+}. This leads to an approximation to the distribution of the z_i, as there is a one-to-one relationship between $\hat{\alpha}_i$ and z_i. This approximation then leads to a conditional normal approximation of $z_i|z_+$, which is used as the instrumental density. As I increases the accuracy of this normal approximation decreases. In their example, they required 1.6 million iterations for the importance sampling estimate to converge. In these cases, a Metropolis algorithm using the normal approximation to $z_i|z_+, z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_I$ as the candidate transition kernel (for i randomly selected) could greatly speed up the algorithm.

7.2 ESUP accept/reject sampling and the MCEM algorithm

ESUP accept/reject sampling (Chapter 5) generates approximately i.i.d. samples by adaptively estimating the supremum needed for exact accept/reject sampling. In practice ESUP rejection sampling accepts essentially the exact same candidates as known sup (KSUP) rejection sampling. In this dissertation we verified this fact theoretically and tested it with several examples. We further show that an infinite supremum may be detected very easily. These ideas coupled

with the Laplace/t candidate nearly fully automate accept/reject sampling. We illustrated a particularly useful application in Chapter 6 where we apply ESUP accept/reject sampling to the MCEM algorithm. We further show how to reduce the Monte Carlo variance of the MCEM algorithm, by simply applying good sampling techniques. An important application of this methodology occurs in categorical response data where counts of observed responses dominate the Monte Carlo variability in the parameter estimates.

A drawback of accept/reject sampling in general is the waste in the unaccepted candidates and the random uniform numbers used in the algorithm. These problems were somewhat addressed by using the Laplace/t candidate distribution, which generally resulted in high acceptance rates. We can further re-use the wasted uniforms as follows. Casella and Robert (1998) argue that, as the uniform random variables are ancillary to the quantity being estimated (the Q function in our case), it is good statistical practice to condition on them. Further, they give an easy formula for their "Rao-Blackwellized" sample mean of accepted candidates. For the MCEM algorithm, this conditioning will reduce the variance of the Q estimate. The interesting question is whether the added computations from "Rao-Blackwellizing" outweigh the gains in Monte Carlo variance reduction for the MCEM algorithm.

ESUP rejection sampling and the empirical Laplace approximation makes sampling from the conditional distribution $f(u_i|\mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t)$ easy (in the notation of Chapter 6). Recall to update σ_t we require an estimate of

$$E_t^*[U_i^2] = \int u_i^2 f(u_i|\mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t) du_i. \tag{7.2}$$

Assuming all constants are known, the ideal Monte Carlo estimate of (7.2) is not the average of squared $f(u_i|\mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t)$ variates. Rather the ideal estimate is to

simulate from the density

$$\tilde{g}(u_i) = u_i^2 f(u_i|\mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t)/E_t^*[U_i^2]$$

and use $\tilde{g}$ as the instrumental density for an importance sampling algorithm. It is easy to see the optimality of this approach as the importance sampling estimate is $E_t^*[U_i^2]$ for every iteration. Of course, as we do not know $E_t^*[U_i^2]$, a practical implementation of this method requires the use of the ratio estimator (2.4). If $U_{i1}, \ldots, U_{in}$ are $\tilde{g}$ variates then the ratio estimate reduces to

$$n(\sum_{j=1}^{n} U_{ij}^{-2})^{-1}.$$

It would be an interesting problem to see if gains could be made by using $\tilde{g}$ in an importance sampling algorithm rather than sampling directly from $f(u_i|\mathbf{Y}_i = \mathbf{y}_i, \boldsymbol{\beta}_t, \sigma_t)$. Though slightly more complex, it would also be possible to repeat this process for the estimates of $E_t^*[\pi_{ij}]$ and $E_t^*[\pi_{ij}(1-\pi_{ij})]$ to update $\boldsymbol{\beta}_t$.

We end the dissertation with the attainable research proposal of incorporating some of the algorithms we developed into general purpose software for fitting models for categorical response data. Ideally, software for performing exact conditional tests for log-linear models would start with Booth and Butler's algorithm then switch to the algorithm from this dissertation when the number of occurrences of tables with negative entries was too high. Other software for fitting binary response GLMMs would ideally offer the option to fit models using: marginal maximum likelihood via numerical quadrature, the EM algorithm with numerical quadrature, the MCEM algorithm with exact accept/reject sampling and the MCEM algorithm with ESUP accept/reject sampling.

REFERENCES

Agresti, A. (1990). *Categorical Data Analysis*, Wiley, New York.

Agresti, A. (1992). A survey of exact inference for contingency tables (with discussion), *Statistical Science* **7**: 131–153.

Agresti, A. (2001). Exact inference for categorical data: Recent advances and continuing corntroversies, *To Appear in Statistics in Medicine*.

Agresti, A. A., Booth, J., Hobert, J. and Caffo, B. S. (2000). Random effects modeling of categorical resonse data, *Sociological Methodology* **30**: 27–80.

Agresti, A., Wackerly, D. and Boyett, J. M. (1979). Exact conditional tests for cross-classifications: Approximation of attained significance levels, *Psychometrika* **44**: 75–84.

Aitkin, M. (1996). A general maximum likelihood analysis of overdispersion in generalized linear models, *Statistics and Computing* **6**: 251–262.

Aitkin, M. (1999). A general maximum likelihood analysis of variance components in generalized linear models, *Biometrics* **55**: 117–128.

Barnard, G. (1945). A new test for 2×2 tables, *Nature* **156**: 177.

Basu, D. (1977). On the elimination of nuisance parameters, *Journal of the American Statistical Association* **72**: 355–366.

Berger, R. and Boos, D. (1994). P values maximized over a confidence set for the nuisance parameter, *Journal of the American Statistical Association* **89**: 1012–1016.

Besag, J. and Clifford, P. (1989). Generalized Monte Carlo significance tests, *Biometrika* **76**: 633–642.

Besag, J., Green, P., Higdon, D. and Mengersen, K. (1995). Bayesian computation and stochastic systems (with discussion), *Statistical Science* **10**: 3–41.

Billingsley, P. (1995). *Probability and Measure*, third edn, Wiley, New York.

Booth, J. and Butler, R. (1999). An importance sampling algorithm for exact conditional test in log-linear models, *Biometrika* **86**: 321–332.

Booth, J. G. and Hobert, J. P. (1999). Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm, *Journal of the Royal Statistical Society, Series B, Methodological* **61**: 265–285.

Booth, J. G. and Sarkar, S. S. (1998). Monte Carlo approximation of bootstrap variances, *The American Statistician* **52**: 354–357.

Boulton, D. M. and Wallace, C. S. (1973). Ocupancy of a rectangular array, *Computer Journal* **16**: 57–63.

Bunea, F. and Besag, J. (2000). MCMC in IxJxK contingency tables, *Fields Institute Communications* **26**: 25–36.

Burrows, P. M. (1979). Selected percentage points of Greenwood's statistic, *Journal of the Royal Statistical Society, Series A, General* **142**: 256–258.

Butler, R. W., Sutton, R. K., Booth, J. G. and Ohman, P. A. (1999). Boosted saddlepoint approximation, *Technical report*, Department of Statistics, University of Florida.

Caffo, B. S., Booth, J. G. and Davison, A. C. (2001). ESUP rejection sampling, *Technical report*, Department of Statistics, University of Florida.

Casella, G. and Berger, R. L. (1990). *Statistical Inference*, Wadsworth, New York.

Casella, G. and Robert, C. P. (1996). Rao-blackwellisation of sampling schemes, *Biometrika* **83**: 81–94.

Casella, G. and Robert, C. P. (1998). Post-processing accept-reject samples: Recycling and rescaling, *Journal of Computational and Graphical Statistics* **7**: 139–157.

Chib, S. and Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm, *American Statistician* **49**: 327–335.

Chow, Y. S. and Teicher, H. (1997). *Probability Theory: Independence, Interchangeability, Martingales*, third edn, Springer-Verlag, New York.

Cochran, W. G. (1977). *Sampling Techniques*, third edn, Wiley, New York.

Coull, B. and Agresti, A. (1999). The use of mixed logit models to reflect subject heterogeneity in capture-recapture studies, *Biometrics* **55**: 294–301.

Cox, D., Little, J. and O'Shea, D. (1997). *Ideals, Varieties, and Algorithms*, Springer-Verlag, New York.

Cox, D. R. and Reid, N. (1987). Parameter orthogonality and approximate conditional inference (with discussion), *Journal of the Royal Statistical Society, Series B, Methodological* **49**: 1–18.

Currie, I. D. (1981). Further percentage points of Greenwood's statistic, *Journal of the Royal Statistical Society, Series A, General* **144**: 360–363.

Daniels, H. E. (1954). Saddlepoint approximation in statistics, *Annals of Mathematical Statistics* **25**: 631–650.

Davison, A. C. (1988). Approximate conditional inference in generalized linear models, *Journal of the Royal Statistical Society, Series B, Methodological* **50**: 445–461.

Davison, A. C. and Smith, R. L. (1990). Models for exceedances over high thresholds (with discussion), *Journal of the Royal Statistical Society, Series B, Methodological* **52**: 393–442.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion), *Journal of the Royal Statistical Society, Series B, Methodological* **39**: 1–38.

Devroye, L. (1986). *Non-uniform Random Variate Generation*, Springer-Verlag, New York.

Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions, *Annals of Statistics* **26**: 363–397.

Feller, W. (1948). On the Kolmogorov-Smirnov limit theorems for empirical distributions, *The Annals of Mathematical Statistics* **19**: 177–189.

Fienberg, S. (2000). Contingency tables and log-linear models: basic results and new developments, *Journal of the American Statistical Association* **95**: 643–647.

Fienberg, S., Makov, U., Meyer, M. and Steel, R. (1999). Computing the exact distribution for a multi-way contingency table conditional on its marginal totals, *Technical report*, Department of Statistics, Carnegie Mellon Univeristy.

Forster, J. J., McDonald, J. W. and Smith, P. W. F. (1996). Monte Carlo exact conditional tests for log-linear and logistic models, *Journal of the Royal Statistical Society, Series B, Methodological* **58**: 445–453.

Gail, M. and Mantel, N. (1977). Counting the number of $r \times c$ contingency tables with fixed margins, *Journal of the American Statistical Association* **72**: 859–862.

Gaver, D. P. and O'Muircheartaigh, I. G. (1987). Robust empirical Bayes analyses of event rates, *Technometrics* **29**: 1–15.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo (with discussion), *Statistical Science* **7**: 473–483.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling, *Applied Statistics* **41**: 337–348.

Good, I. J. (1976). On the application of symmetric Dirichlet distributions and their mixtures to contingency tables, *The Annals of Statistics* **4**: 1159–1189.

Good, I. J. (1979). A comparison of some statistical estimates for the numbers of contingency tables, *Journal of Statistical Computation and Simulation* **8**: 312–314.

Greenland, S. (1991). On the logical justification of conditional tests for two-by-two contingency tables (c/r: 92v46 p163), *The American Statistician* **45**: 248–251.

Guo, S. W. and Thompson, E. A. (1992). Performing the exact test of Hardy-Weinberg proportion for multiple alleles, *Biometrics* **48**: 361–372.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**: 97–109.

Hirji, K. F. (1996). A note on exact analysis of several 2×2 tables, *Biometrics* **52**: 1018–1025.

Hirji, K. F., Mehta, C. R. and Patel, N. R. (1987). Computing distributions for exact logistic regression, *Journal of the American Statistical Association* **82**: 1110–1117.

Hirji, K. F., Tsiatis, A. A. and Mehta, C. R. (1989). Median unbiased estimation for binary data, *The American Statistician* **43**: 7–11.

Landis, J. R. and Koch, G. G. (1977). An application of hierarchical Kappa-type statistics in the assessment of majority agreement among multiple observers, *Biometrics* **33**: 363–374.

Lauritzen, S. L. (1996). *Graphical Models*, Oxford University Press, London.

L'Ecuyer, P. (1997). Tables of maximally-equidistributed combined lsfr generators, *Technical report*, University of Montral, Canada.

Lugannani, R. and Rice, S. O. (1980). Saddlepoint approximation for the distribution of the sum of independent random variables, *Advances in Applied Probability* **12**: 475–490.

Marsaglia, G. and Zaman, A. (1994). Some portable very-long-period random number generators, *Computers in Physics* **8**: 117–121.

McCullagh, P. (1986). The conditional distribution of goodness-of-fit statistics for discrete data, *Journal of the American Statistical Association* **81**: 104–107.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models*, second edn, Chapman & Hall, London.

McCulloch, C. E. (1997). Maximum likelihood algorithms for generalized linear mixed models, *Journal of the American Statistical Association* **92**: 162–170.

McDonald, J. W., Smith, P. W. F. and Forster, J. J. (1999). Exact tests of goodness of fit of log-linear models for rates, *Biometrics* **55**: 620–624.

Mehta, C., Patel, N. and Senchauduri, P. (2000). Efficient Monte Carlo methods for conditional logistic regression, *Journal of the American Statistical Association.*

Mehta, C. R. and Patel, N. R. (1980). A network algorithm for the exact treatment of the $2 \times k$ contingency table, *Communications in Statistics, Part B – Simulation and Computation* **9**: 649–664.

Mehta, C. R. and Patel, N. R. (1983). A network algorithm for performing Fisher's exact test in $r \times c$ contingency tables, *Journal of the American Statistical Association* **78**: 427–434.

Mehta, C. R. and Patel, N. R. (1995). Exact logistic regression: Theory and examples, *Statistics in Medicine* **14**: 2143–2160.

Mykland, P., Tierney, L. and Yu, B. (1995). Regeneration in Markov chain samplers, *Journal of the American Statistical Association* **90**: 233–241.

Pagano, M. and Tritchler, D. (1983). On obtaining permutation distributions in polynomial time, *Journal of the American Statistical Association* **78**: 435–440.

Park, S. and Miller, K. (1988). Random number generators: good ones are hard to find, *Communications of the ACM* **31**: 1192–1201.

Patefield, W. M. (1981). [algorithm As 159] An efficient method of generating random $r \times C$ tables with given row and column totals, *Applied Statistics* **30**: 91–97.

Patel, N., Mehta, C. R. and Senchauduri, P. (1999). Markov chain Monte Carlo exact inference for exact logistic regression, Invited Session, Joint Statistical Meetings, Baltimore.

Paul, S. and Deng, D. (2000). Goodness of fit of generalized linear models to sparse data, *Journal of the Royal Statistical Society, Series B, Methodological* **62**: 323–333.

Pickands, James, I. (1975). Statistical inference using extreme order statistics, *The Annals of Statistics* **3**: 119–131.

Pierce, D. A. and Peters, D. (1992). Practical use of higher order asymptotics for multiparameter exponential families (with discussion), *Journal of the Royal Statistical Society, Series B, Methodological* **54**: 701–725.

Pierce, D. A. and Peters, D. (1999). Improving on exact tests by approximate conditioning, *Biometrika* **86**: 265–277.

Presnell, B. (1996). Bootstrap unconditional p-values for the sign test with ties and the 2×2, *Journal of Nonparametric Statistics* **7**: 47–55.

Pyke, R. (1965). Spacings (with discussion), *Journal of the Royal Statistical Society, Series B, Methodological* **27**: 395–449.

Read, T. R. C. and Cressie, N. A. C. (1988). *Goodness-of-fit Statistics for Discrete Multivariate Data*, Springer-Verlag, New York.

Reid, N. (1995). The roles of conditioning in inference (with discussion), *Statistical Science* **10**: 138–157.

Robert, C. P. and Casella, G. (1999). *Monte Carlo Statistical Methods*, Springer, New York.

Routledge, R. D. (1992). Resolving the conflict over Fisher's exact test, *The Canadian Journal of Statistics* **20**: 201–209.

Searle, S. R. (1997). *Linear Models*, Wiley, New York.

Sen, P. K. and Singer, J. M. (1993). *Large Sample Methods in Statistics*, Chapman and Hall, London.

Smith, P. W. F., Forster, J. J. and McDonald, J. W. (1996a). Monte Carlo exact tests for square contingency tables, *Journal of the Royal Statistical Society, Series A, General* **159**: 309–321.

Smith, P. W. F., McDonald, J. W., Forster, J. J. and Berrington, A. M. (1996b). Monte Carlo exact methods used for analysing interethnic unions in Great Britain, *Applied Statistics* **45**: 191–202.

Smith, R. L. (1985). Maximum likelihood estimation in a class of non-regular cases, *Biometrika* **72**: 67–92.

Stephens, M. A. (1981). Further percentage points for Greenwood's statistic, *Journal of the Royal Statistical Society, Series A, General* **144**: 364–366.

Strawderman, R. L. and Wells, M. T. (1998). Approximately exact inference for the common odds ratio, *Journal of the American Statistical Association* **93**: 1294–1307.

Suissa, S. and Shuster, J. J. (1984). Are uniformly most powerful unbiased tests really best?, *The American Statistician* **38**: 204–206.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion), *The Annals of Statistics* **22**: 1701–1728.

Tierney, L., Kass, R. E. and Kadane, J. B. (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions, *Journal of the American Statistical Association* **84**: 710–716.

Upton, G. J. G. (1982). A comparison of alternative tests for the 2×2 comparative trial, *Journal of the Royal Statistical Society, Series A, General* **145**: 86–105.

Waterman, R. P. and Lindsay, B. G. (1996). A simple and accurate method for approximate conditional inference applied to exponential family models, *Journal of the Royal Statistical Society, Series B, Methodological* **58**: 177–188.

Wei, G. C. G. and Tanner, M. A. (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms, *Journal of the American Statistical Association* **85**: 699–704.

Wild, P. and Gilks, W. R. (1993). Algorithm As 287: Adaptive rejection sampling from log-concave density functions, *Applied Statistics* **42**: 701–708.

Yao, Q. and Tritchler, D. (1993). An exact analysis of conditional independence in several 2 × 2 contingency tables, *Biometrics* **49**: 233–236.

Yates, F. (1984). Tests of significance for 2 × 2 contingency tables (with discussion), *Journal of the Royal Statistical Society, Series A, General* **147**: 426–463.

Zelen, M. (1971). The analysis of several 2 × 2 contingency tables, *Biometrika* **58**: 129–137.

BIOGRAPHICAL SKETCH

Brian was born in Colorado Springs, Colorado, on November 28th, 1972. His father, Ronald, mother, Deborah, and older brother, Stephen, lived in several states including Ohio, Alabama and New Jersey until winding up in Jacksonville, Florida in 1985. Brian graduated high school from the Bolles school in Jacksonville in May of 1991. In 1991 Brian enrolled at the University of Florida as a freshman art student and varsity athlete in swimming. In his second year he met his fiance, Kerri, through mutual friend Eric Hoag.

Performing poorly in art classes, Brian switched to mathematics in 1993. Brian graduated with a Bachelor of Science with honors majoring in mathematics and statistics in December of 1995. A summer of working at the (then) Pediatric Oncology Group Statistical Office with constant encouragement by POG co-principal investigator Dr. Jim Kepner convinced Brian to attend statistics graduate school at the University of Florida over a career in pure mathematics.

In his second year of graduate school Professor James Booth encouraged Brian to consider pursuing a PhD. After an independent study course with Dr. Booth, Brian decided to work towards a PhD under his supervision. In his third year Brian was fortunate to receive a research assistantship to work with categorical data analysis guru Dr. Alan Agresti.

On April 21st, 2001, Brian formally proposed to Kerri in Baltimore. Around the same time, Brian accepted a position as an assistant professor at the Johns Hopkins University Department of Biostatistics, shortly after Kerri matched with the medical residency program at the University of Maryland Hospital in Baltimore, Maryland.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

James G. Booth

James G. Booth, Chair
Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Alan Agresti
Distinguished Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

James Hobert
Associate Professor of Statistics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Scott McCullough
Associate Professor of Mathematics

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Andrew Rosalsky
Professor of Statistics

This dissertation was submitted to the Graduate Faculty of the Department of Statistics in the College of Liberal Arts and Sciences and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 2001

Dean, Graduate School